DOCUMENT RESUME

ED 352 033 IR 054 254

AUTHOR	Lancaster, F. W., Ed.; Smith, Linda C., Ed.
TITLE	Artificial Intelligence and Expert Systems: Will They Change the Library? Papers Presented at the Annual Clinic on Library Applications of Data Processing (27th, Urbana, Illinois, March 25-27, 1990). Illinois, March 25-27, 1990).
INSTITUTION	Illinois Univ., Urbana. Graduate School of Library and Information Science.
REPORT NO	ISBN-0-87845-084-X; ISSN-0069-4789
PUB DATE	92
NOTE	303p.; For the proceedings of the 1986 Processing Clinics, see ED 293 562.
PUB TYPE	Collected Works - Conference Proceedings (021) -- Books (010)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Artificial Intelligence; Cataloging; *Expert Systems; Futures (of Society); *Indexing; *Information Management; *Information Technology; Library Automation; Library Collection Development; Library Technical Processes; Online Searching; Reference Services; Users (Information)
IDENTIFIERS	Document Delivery; *Knowledge Representation; *Natural Language Processing

ABSTRACT

Some of the 12 conference papers presented in this proceedings focus on the present and potential capabilities of artificial intelligence and expert systems as they relate to a wide range of library applications, including descriptive cataloging, technical services, collection development, subject indexing, reference services, database searching, and document delivery. Other papers deal with the underlying design issues of knowledge representation and natural language processing. The papers are; (1) "Artificial Intelligence: What Will They Think of Next?" (Douglas P. Metzler); (2) "Technical Services Processes as Models for Assessing Expert System Suitability and Benefits" (Charles Fenly); (3) "Automated Cataloging: Implications for Libraries and Patrons" (Stuart Weibel); (4) "Interactive Knowledge-Based Systems for Improved Subject Analysis and Retrieval" (Susanne M. Humphrey); (5) "Reference Expert Systems: Foundations in Reference Theory" (James R. Parrott); (6) "Expert Systems at the National Agricultural Library: Past, Present, and Future" (Samuel T. Waters); (7) "User Models for Information Systems: Prospects and Problems" (Christine L. Borgman and Yolanda I. Plute); (8) "Natural Language Processing: Current Status for Libraries" (Amy Warner); (9) "Knowledge Representation in Artificial Intelligence" (Irene L. Travis); (10) "Intelligent Interfaces to Online Databases" (Brian C. Vickery); (11) "Expert Systems in Document Delivery: The Feasibility of Learning Capabilities" (Jaime Pontigo, Ezequiel Tovar-Reyes, Guillermo Rodriquez, and Sergio Ortiz-Gama); and (12) "Walking Your Talk: Why Information Managers Are Not High Tech" (W. David Penniman). An index and brief author biographies conclude the volume; chapters include references. (KRN)

Artificial Intelligence and Expert Systems: Will They Change the Library?

Papers presented at the 1990 Clinic on Library Applications
of Data Processing, March 25-27, 1990

Clinic on Library Applications of Data Processing: 1990

Artificial Intelligence and Expert Systems: Will They Change the Library?

Edited by
F. W. LANCASTER
and
LINDA C. SMITH

Graduate School of Library and Information Science
University of Illinois at Urbana-Champaign

© 1992 by The Board of Trustees of the University of Illinois
ISBN 0-87845-084-X ISSN 0069-4789

Printed in the United States of America
on acid-free paper

CONTENTS

Introduction 1
F. W. Lancaster and Linda C. Smith

Artificial Intelligence:
What Will They Think of Next? 2
Douglas P. Metzler

Technical Services Processes as Models for Assessing
Expert System Suitability and Benefits 50
Charles Fenly

Automated Cataloging: Implications for Libraries
and Patrons 67
Stuart Weibel

Interactive Knowledge-Based Systems for Improved Subject
Analysis and Retrieval 81
Susanne M. Humphrey

Reference Expert Systems: Foundations in Reference Theory 118
James R. Parrott

Expert Systems at the National Agricultural Library:
Past, Present, and Future 161
Samuel T. Waters

User Models for Information Systems: Prospects and Problems 178
Christine L. Borgman
Yolanda I. Plute

Natural Language Processing: Current Status for Libraries 194
Amy Warner

Knowledge Representation in Artificial Intelligence 215
Irene L. Travis

Intelligent Interfaces to Online Databases 239
Brian C. Vickery

Contents *(Cont.)*

Expert Systems in Document Delivery: The Feasibility of Learning Capabilities 254
Jaime Pontigo
Ezequiel Tovar-Reyes
Guillermo Rodriguez
Sergio Ortiz-Gama

Walking Your Talk: Why Information Managers are Not High Tech 267
W. David Penniman

Contributors 278

Index 283

Introduction

"Artificial intelligence" and "expert systems" are terms that have appeared with increasing frequency in the literature of library and information science. Some writers have been cautious in their claims, but others have been rather extravagant, implying the existence of capabilities well beyond those of systems that now exist or are likely to in the immediate future. Misuse of the terminology has also occurred in the literature; in particular, the term "artificial intelligence" has been applied to techniques that involve computation but no real intelligence.

The 27th Annual Clinic on Library Applications of Data Processing, held March 25-27, 1990, at the University of Illinois at Urbana-Champaign, was designed to correct some of these misconceptions by presenting a balanced picture of present and potential capabilities of artificial intelligence and expert systems. The papers presented here deal with these capabilities as they relate to a wide range of library applications: descriptive cataloging, technical services, collection development, subject indexing, reference services, database searching, and document delivery. Other papers deal with the underlying design issues of knowledge representation and natural language processing.

We hope that this volume will be of interest and value to all those wanting a better understanding of the possibilities and problems associated with the application of artificial intelligence/expert systems in libraries.

F. W. LANCASTER
LINDA C. SMITH
Editors

DOUGLAS P. METZLER

Department of Information Science
University of Pittsburgh

Artificial Intelligence: What Will They Think of Next?

ABSTRACT

This paper explores several points regarding the development of the field of artificial intelligence and its potential impact on library science. The discussion is motivated by the nature of future library collections and services that will be made available (in part) through artificial intelligence applications and by the basic need for intelligent analysis of the vast volumes of data and information that will be available through continuing developments in storage and communication technology. The general concept of intelligence is shown to involve a number of more specific types of thinking, and dimensions and objects of thought, and several examples of current research areas concerned with these more specific problems are described. A general fieldwide dialectic between research oriented towards these specific problems and research oriented towards the integration of these specific capabilities into broader systems is described and related to the general question of improving the capabilities of interactive intelligent systems. These issues are discussed in the context of several definitions of intelligence and artificial intelligence and are illustrated in the example of a specific system.

INTRODUCTION

This conference began twenty-six years ago to explore applications of computer technology, often called "data processing" at the time, to libraries. This year, the conference will explore the potential impact on libraries of what is in many respects the most advanced aspect of computational technology, that of artificial intelligence and the particular subarea within it known as expert systems. Where do we stand in the field of artificial intelligence? In the last several years, there has been a dramatic increase in the level of activity related to library applications of artificial intelligence, much of it conducted by the participants at this conference. And yet, even given these efforts, it is useful to begin this process by asking what the general prospects are for artificial intelligence applications in the domain of library science. One reason for this concern is simply the fact that libraries constitute, because of the size and variety of materials that they contain, an extremely difficult problem for any technological application. This is evidenced by the relatively slow pace at which even conventional technologies and capabilities such as online catalogs and compact disk readers have moved into widespread use. The deeper library applications of artificial intelligence will be faced in a much more direct way with the breadth and depth of the contents of a library, than has been any other technological facet of librarianship.

This review will begin by suggesting a general scenario of what the library of the future might look like, and how artificial intelligence will influence the nature of the collections and services that will be possible. Next, it will present a characterization of the importance of intelligent analysis for any efficient utilization of the information resources of the future. The basic point is that intelligent analysis is inevitable if we are to cope with the information explosion provided by the communication and storage media of the future. To put it another way, even the tremendous storage and transmission facilities that will be available will be inadequate for what might seem to be fairly straightforward applications if they are not handled in ways that are based on intelligent analysis.

Assuming that intelligent analysis is necessary, we will next turn to ask the question, "What will it take to accomplish this?" There are, of course, many aspects to the answer to that question; fortunately, two of the most important general aspects of this question, the areas of knowledge representation and natural language processing, will be reviewed separately later in this conference, and so will not need detailing here. The emphasis will be on the question of what kinds of thinking processes do people seem to employ, and how successfully can we deal with these processes in artificial intelligence? This discussion is organized around the following definitions of intelligence and artificial intelligence, each of which contributes to a general sense of what the field of artificial intelligence is about, including what the difficulties are and why it proceeds in the manner in which it does:

1. Intelligence is data, information, and knowledge compression
2. Intelligence is a shared body of knowledge
3. Intelligence is appropriate action
4. Intelligence is entailment
5. Intelligence is common sense reasoning
6. Intelligence is culture, communication, and cooperation
7. Intelligence is the ability to learn

In the course of discussing these definitions, particularly the latter three, several key general research areas that are currently being explored in artificial intelligence will be introduced, and an attempt will be made to show how these abstract problems relate to the problem of developing general artificial intelligence capabilities. In developing the implications of these definitions of intelligence, several themes regarding the nature of artificial intelligence and how research is proceeding in this area will also be explored. These include:

1. The need to simulate human thinking. Although some people in artificial intelligence question the need to simulate human thinking, (i.e., "airplanes don't flap their wings"), there is a growing realization that some aspects of human thinking are critical for any successful intelligent system. For instance, it seems necessary in scientific or engineering problem solving to approach problems first in a qualitative "common sense" manner prior to bringing full quantitative rigor to bear, if for no other reason than computational efficiency. Furthermore, for systems to engage in scientific discovery in these sorts of domains, they need to incorporate ideas relating to the "interestingness" of concepts, and this also takes us outside the realm of the narrowly defined science itself and seems to introduce elements of human cognition. In any case, for those of us concerned with library applications of artificial intelligence, there is little choice but to embrace the cognitively oriented approach to artificial intelligence, since the majority of applications we are interested in will involve analyzing and utilizing the vast store of the products of human thought that is contained in libraries, and mediating between those products and the thought processes of users. The ability to deal intelligently with the vast scope of the contents of the libraries of the world is by far the most difficult challenge that one can pose for artificial intelligence and natural language processing, although, of course, as in all application domains, there are many useful smaller and shorter-term projects that can be undertaken.
2. The dialectics of artificial intelligence. A second theme regarding the nature of artificial intelligence will concern the manner in which the overall research program seems to employ a dialectic between research concerned with analyzing particular issues in great depth

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

- Points of view or opinions stated in this document do not necessarily represent official OERI position or policy

ED352033

12054254

BEST COPY AVAILABLE

"PERMISSION TO REPRODUCE THIS MATERIAL IN OTHER THAN PAPER COPY HAS BEEN GRANTED BY
H. Wagner

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

and research concerned with synthesizing some of the results of such work in larger systems that cut across a number of issues.

3. Learning: A critical issue but not a panacea. One of the particular issues in artificial intelligence is that of machine learning. While the importance of this issue in the general scheme of things in artificial intelligence should be stressed, one should guard against the temptation, which has hundreds of years of precedent in Western thought, to try to avoid a direct assault on all of the other tough issues regarding the nature of thought by simply attempting to build a learning machine.
4. Requirements for intelligent interactive systems. Having looked at a sample of the abstract problems involved in simulating thinking, this paper will turn to expert systems—what is required to build more useful intelligent systems applications, particularly interactive systems, and how these requirements relate to the sorts of abstract problems already discussed. A system under development at the University of Pittsburgh, whose purpose is to aid in the design and diagnosis of local area networks, will illustrate these issues.
5. What is artificial intelligence? This paper concludes with some general considerations of the fundamental nature of artificial intelligence; briefly considers the question of the need for intelligent systems to be embedded in the world, and whether that curtails the promise of really intelligent artificial systems; and then turns to one final definition of artificial intelligence that seems to avoid these sorts of problems and nicely summarizes the potential of artificial intelligence to change not only our libraries but our entire civilization.

THE LIBRARY OF THE FUTURE

The impact of advanced computer technology in general, and artificial intelligence and expert systems in particular, on the nature of the library of the future will be immense and qualitatively quite different even from what we are anticipating now with our current work. Most of the library-oriented expert systems and artificial intelligence applications which have been developed to date, or which are currently under development, are essentially aids to facilitate the business of running libraries as they are structured today. These applications and potential applications include systems to aid in carrying out the support operations of the library, such as collection development; budget, personnel, and scheduling arrangements; and disaster planning and response. They also include systems to enhance user services (or to off-load some of the more routine aspects of user services) in areas such as ready reference and information retrieval.

Conventional computer-based approaches are available or possible for some of these sorts of applications, and artificial intelligence has been employed in these applications in a variety of ways including projects which use artificial intelligence as an enhancement for conventional approaches, and work which uses artificial intelligence programming techniques just to improve the ease with which conventional programs (e.g., decision trees) can be implemented. Other projects involve work which inherently does require artificial intelligence technology for one or more of the usual reasons (e.g., the combinatorial explosion of possible alternative situations, the need to deal with uncertainty, the need to provide transparent flow of control, the need to capture human heuristics, etc.). Taken together, these sorts of applications are interesting and important from any of several perspectives, but they will not, in themselves, change the fundamental nature of the library, except to make it more useful and more economical to run.

Fundamental changes in the nature of the library will occur as a result of changes in the nature of the materials that are collected. Libraries are the major repositories of the knowledge of the world, but the format in which much of this knowledge is being recorded has begun to undergo profound changes, and the potential uses that can be made of this newly formatted knowledge are changing as well. Artificial intelligence represents a potentially far more profound change in the format and utilization of knowledge. The challenge to librarians is to understand the nature of this change, and to conceptualize the library as the general repository of knowledge, rather than only a repository for books and other static media. If libraries avoid this expanded role, new institutions (electronic bulletin board services, the phone companies, cable TV, CD companies, software manufacturers, etc.) will step in to fill the need, and will inevitably relegate libraries to a much less significant position in society than they now occupy.

Electronic Collections of the Future

What will the future library contain and what will it be able to provide for the user based on these materials? This section will provide a range of possibilities, ranging from the immediately do-able to the long-range possibilities. It can be considered a research blueprint for the library of the future (Cerf, 1989).

Documents

Electronic documents can be printed by anyone who has access to a printer, and they can be tailored in various ways to the individual needs of the user. (The legal and financial issues thus raised are outside the scope of this paper.) As documents become more frequently prepared in digital form, and as advantage begins to be taken of the potential that this provides, it will be possible to package, with the document, information regarding how it can best be presented to, and used by, the user. At the least this will involve information concerning how the document can be printed on different devices. It will also be possible to provide active indexing or glossary facilities, perhaps even hypertext-like progressive deepening, according to the interest and expertise of the reader. Some of the library-level indexing (e.g., to related books of interest) could also be provided within a given text; and could potentially be tailored to the interests of individual users. (Questions of authorship and control of the presentation of intellectual property that these capabilities imply are, again, outside the scope of this paper.)

Data

Both conventional databases and large collections of "raw" sensory data (e.g., seismic wave recordings, or recordings returned by satellites or space probes) could be made available online. These could be designed to provide the appropriate "views" for users of various types. This sort of interpretation would be far more difficult in the case of raw data, which is hardly ever useful in an uninterpreted state. (In the short term, of course, this problem of interpretation could be left to the user, but this would certainly limit the use of the information.)

Programs and Knowledge Bases as Commodities

Like current electronic bulletin boards, the library of the future will provide users with various sorts of programs and materials to develop and/or utilize programs. These could include standard applications as well as expert systems. A particularly interesting set of such applications, considering the major role in education that libraries have always taken, would be computer-aided instruction programs and intelligent-tutoring systems. Since expert systems and many other forms of artificial intelligence provide a high degree of separation between the domain-specific knowledge (both conceptual, as in inheritance hierarchies, and procedural, as in production rules) and the system that utilizes that knowledge, it will also be very useful to begin providing libraries of such knowledge bases, to provide support for future system development.

Procedural Access to the Knowledge

The library of the future may be able to provide a far richer access to, and utilization of, the knowledge contained (often implicitly) in its collection. The most immediately feasible development along this line would be content-based information retrieval. This, of course, would require a far more general and robust brand of artificial intelligence and natural language understanding than what we have available now. The step beyond that would involve not only understanding text well enough to determine whether it is relevant to a general information need expressed by a user, but to understand it well enough to actually extract information that can be used by a program. There are difficulties here, since even textbooks normally provide only a declarative description of a domain, rather than a procedural description of how to carry out the relevant activities in that domain, but there are potential ways out of this dilemma. For instance, the book might provide declarative background for an intelligent workbench intended for a knowledgeable user. Thus, a skilled designer might use the system to check that he or she is using unfamiliar materials in ways that conform to the limits of the materials.

The Physical Library of the Future

Electronic media do not of themselves require the current physical structure of a library; moreover, there is no reason to restrict a user's access to what is locally available. So one vision of the library of the future consists of a vast distributed network of data, information, and knowledge, accessed via intelligent information gathering and synthesizing agents that traverse the network in service of users' needs (Cerf, 1989). This scenario, only hinted at by today's research networks, bulletin boards, and distributed information retrieval facilities, would offer immense power to individual users but will require major advances in a number of technologies to implement in a full sense.

In any case, the physical library certainly will not disappear in the foreseeable future, although it might change in some very dramatic ways. First, of course, we will still have media such as books, film, and art and the need to continue the traditional role of libraries. Second, the new electronic technologies will themselves create several legal and economic issues that can probably best be dealt with in terms of a setting such as that of the physical library. For instance, individuals and even institutions cannot afford to purchase all of the software and hardware (including special peripheral devices) they might occasionally need. As various databases, knowledge bases, and intelligent systems become more openly available, this situation will become more apparent, and it will become necessary to develop some sort of distribution mechanism that can make these resources temporarily available to the general public and at the same time permit manufacturers and developers to maintain economic viability (note that the concept of the free public library may require some modification in this process). The library of the future may develop, in part, to resemble the campus computing center, which is more and more growing to resemble a lending library of computing resources rather than simply an access point to a central computer. Finally, the library of the future will probably still provide a good deal of its functionality in physical structures similar to today's simply because it is a useful and satisfying social institution. On today's campuses, computing centers and libraries are both the foci of much useful interaction and mutual learning, and there is no reason to think that similar facilities would not be as useful and enjoyable for the general public. In addition, the public is likely to become more willing to invest public funds in such resources as its investment in personal computing increases.

Why Will This Happen?

Definition 1
Intelligence is data, information, and knowledge compression

The scenario portrayed in the last section (or something very much like it) will come about because it will be driven by the necessity to deal adequately with the expectations and demands created by the increasing availability of resources such as: CD and other vast personal storage media, wide area network access, fiber optic connections to local work points and homes, and powerful personal computers and workstations. Artificial intelligence is the only way to deal with this information explosion.

The reason for this is that dealing with the world through "brute force" data analysis is doomed to failure. A single screen image on a typical 19-inch high-resolution monitor involves a million pixels, each of which can take several bits, depending on the color or gray scales used in its representation. One second of high-definition TV requires 4MG of data, which is only four times greater than that required even by the rather poor resolution of standard TV (Lucky, 1989). (This is why one cannot simply digitize the incoming signal, store it, and then "mouse" back to earlier points in the transmission for an instant replay, the way one can in the windowing systems of computers.) Although these sorts of transmission rates are on the edge of what is feasible for general fiber optic technology, they will clearly overwhelm by orders of magnitude any foreseeable attempts to store large amounts of the transmissions, utilize multiple simultaneous channels, or otherwise manipulate the data.

The limits of brute force storage of unanalyzed data are nicely illustrated by the Library of Congress project designed to preserve copies of the millions of books which are rapidly degenerating. Since much of the value of these books lies in their original appearance, and not just their content, accurate reproduction of these objects would take billions of bits per page. Consequently, the project is using photography (microfiche) rather than digital analysis, in spite of the inherent limits of retrievability and reproductive fidelity that the analog medium implies (Battin, 1989).

If the raw sensory data made available by the world requires such an enormous amount of storage, how was it possible for intelligent life to evolve and function? The answer is clearly that intelligent life (meaning life capable of responding appropriately to its environment) does not require the storage of vast quantities of unanalyzed raw sensory data, but rather requires a selective sensitivity to various important properties of that data, and the recording of only a small portion even of those properties to which it is initially sensitive. For instance, our human representation (memory) of visual experience is more like a description (i.e., an abstract summary of the most important features of an experience) than like a photographic record, even though it does record a good deal of analog information. In a word, we respond to and store "content," i.e., information, not raw data. Since these interpretations require less storage (or transmission) capacity than does the original raw data, intelligence can be viewed as involving a series of "data compression" operations that summarize and extract from direct experience those aspects of the world which are most relevant to the intelligent system itself.

Definition 2
Intelligence is a shared body of knowledge

There is another observation that makes this point. The more that is known about something, the less information is required to notice, remember, or convey any particular information in that area. For instance, it is extremely difficult to remember a sentence in an unknown language, but it becomes progresssively simpler if: (a) one knows the language, (b) one knows about the general topic of the sentence, and (c) one knows about the specific items mentioned in the sentence. Similarly, but on a smaller scale, when text is communicated digitally, it is almost never sent directly as bitmaps of the individual letters (although some computers do this locally in driving a printer). Rather, a code such as ASCII is used to indicate (using only a few bits) which letter is intended, and the host systems share knowledge of how to print that character. If several fonts are available, they can be indicated by communicating only a few bits whenever the fonts switch, as long as, once again, the systems share their knowledge of the fonts. The goal of natural language understanding can be viewed in this context as the attempt to isolate or extract the various sources of knowledge, both linguistic, i.e., phonological/graphemic, morphological, syntactic, semantic and pragmatic, and general world knowledge, which are shared by the speakers of a language, and which form the background within which new linguistic utterances are interpreted. Since these are already shared, they need not be communicated, and the amount of new information conveyed by an utterance is actually quite limited. There is evidence that this is how people communicate. For instance, after only a few minutes, all people can remember is the gist, not the form, of the utterances they have heard or read. One goal of designing intelligent computational systems is to take advantage of this same principle of shared knowledge, to enable them to economically store and transmit the gist of messages.

Libraries provide an extremely difficult challenge for this undertaking since their content is so broad, and the uses to which a user might want to put the collection are equally broad themselves. In fact, one could hardly look for a better illustration of the range of capabilities needed for a full artificial intelligence or natural language processing system than that of full-text understanding and content-based retrieval and utilization.

THE COMPLEXITY OF THE WORLD

The previous section argued that efficient use of the vast computational, communications, and storage facilities that are rapidly becoming available will require intelligent analysis of the material used rather than the manipulation of raw data. It also suggested that part of what is required for a general understanding capability is a background body of general world knowledge. The need to encode both this general background knowledge and any specific knowledge such as that contained in documents or communications with users leads to the general question of how to represent that knowledge, and even to the more general questions of what is intelligence and what is knowledge.

Symbolic Representation and the Knowledge Representation Hypothesis

Definition 3
Intelligence is the ability to act appropriately

Intelligence can be defined as *the ability of an organism or agent to carry out appropriate actions in its environment*, where the definition of "appropriate" is very much dependent on the individual organism or agent and the environmental context in which it is situated. One very general approach to defining "appropriate" can be based on the concept of actions that lead to survival. This approach has the advantage of being objectively definable even for very simple organisms with very limited information processing capabilities. The more interesting case for artificial intelligence, however, can be defined in terms of the "appropriateness" of the match between certain internal states which drive the actions of the system in question (i.e., "goals"), and those aspects of the environment to which the organism or agent is sensitive. In these complex information processing situations (especially in the case of human cognition), the organism is not only directly sensitive to immediately given properties of the raw sensory data (e.g., heat and cold, chemical gradients in the air, colors, simple shapes, etc.), it is also, for the most part, sensitive to higher level abstractions and to the implications that environmental objects have for various sorts of activities. In short, the environment as well as the goals of the organism must be interpreted and internally represented in order to provide the substance of cognition. This idea has been dubbed the Knowledge Representation Hypothesis (Smith, 1982). It essentially states that any intelligent system must be composed of elements that: (1) are meaningfully interpretable (more specifically, propositionally interpretable) by an outside observer (which is possible if and only if the elements designate aspects of the world that correspond to how the outside observer can interpret the world); and (2) can simultaneously play a causal role in how the system operates. This idea was originally developed by Newell and Simon in the 1950s. It was at that time the basis for the first successful artificial intelligence programs ever written, and it has been the basis for almost all artificial intelligence programs that have been written since. This idea was developed into the Physical Symbol System Hypothesis (Newell & Simon, 1976), which suggests that any intelligent system will of necessity be based on this architecture. Many properties of human cognition appear to demand the computational mechanisms provided by the symbolic representation hypothesis (Metzler, 1990). In brief, symbols provide a way to internally manipulate relevant aspects of the world. For instance, they permit one to combine symbolic representations to produce new products, as in building the (possibly novel) meaning of a sentence from the meanings of the words, or as in putting together a new plan or design out of the fragments of previous ideas. Similarly, they provide a way of re-presenting aspects of internal or external experience at later points in time, which is also a critical ingredient of higher thinking.

All this seems to lead to a very straightforward program for artificial intelligence. All we need to do is figure out how to represent (in the computer) the symbolic representations contained in the mind. Over the last two decades, a number of representational devices have been developed that seem to do a good job of this. Frame representations are a powerful way of representing concepts, for instance, and rule-based representations seem to capture a good deal of the conditional or "if-then" flavor of thinking. This author dubs the similarity relationship between these representation schemes and the presumed human cognitive representations that they are attempting to capture "Second-Order Platonism" (Metzler, 1990). That is, as Plato sought to explain human semantics by mapping it to an ideal world, we will explain machine semantics simply by mapping it to human semantics (without dealing with Plato's original problem). With it, the deep philosophical questions of machine semantics are avoided by building representations that are to correspond as closely as possible to the basic elements of human thought. (It should be noted, however, that this avoids only the philosophical problems of the "meaning of machine meaning," and is only a first step towards solving the general software engineering problems involved in grasping what a program is doing; it certainly does not avoid the problems of developing clear and precise notions of the semantics of programming constructs so that we can tell exactly how a program ought to behave under specifiable conditions. In his 1990 paper, this author argues, in support of the knowledge representation hypothesis, that this semantic correspondence is necessary for any intelligent system not only (as in the physical symbol system hypothesis) because of the internal requirements of computation, but also because it is the only coherent way in which a large-scale knowledge engineering or software engineering project can be undertaken—i.e., the modules of any large-scale project must be meaningfully interpretable in the world, and the products that the modules transfer among themselves must also be interpretable as corresponding to meaningful aspects of the world. In order to see why the general problem of artificial intelligence is so hard in spite of this Second Order Platonism, it is useful to look at intelligence from yet another perspective.

Entailment

Definition 4

Intelligence is the ability to derive entailments

Another way to conceptualize the general problem of intelligent behavior is in terms of "entailment," the problem of determining what is implicit in a knowledge base. For instance, even the very general problem of deciding what to do next at any particular point in time can be construed as the problem of determining what action, external or internal, is suggested (i.e., entailed) by a system's general knowledge base containing all of the agent's general knowledge of the world, its own goal structures, and its knowledge of the present relevant circumstances of the world. For instance, a knowledge base containing something like the following could be taken to imply something like "establish the goal to Eat (x)" or that the actual action itself should be taken.

```
Goal-"Eat-Something"
Present (Small-fish-0089)
Instance (Small-fish-0089, Fish-type-005)
Edible (Fish-type-005)
If Goal-"Eat-Something"
   and Present(x)
   and Edible(x)
Then Eat(x)
```

Entailment comes in a variety of guises. For instance, if one knows that "John walked to the store," one also knows that "John went to the store," because the meaning of "go" is more general than that of "walk" and thus subsumes it. In addition, one can also infer that John was at the store for some (possibly very short) period of time after this act, he was not at the store prior to the act, he had some sort of reason to go to the store, and he was in reasonably good health when he carried out the action. (Note that the latter rather boring inference becomes much more problematic and hence interesting if we also know that John had just had a serious accident and was in the hospital just prior to this event.) In addition to these sorts of entailments which seem to be based for the most part on the semantics of how we use language, there are entailments that have more to do with the nature of the things in the world per se. For instance, we know that if a gun's trigger is pulled, and the gun is loaded, it will fire. It is hard to determine exactly where to make this distinction between lexical/linguistic and general world knowledge, however.

Formal (and informal) computational procedures exist for *deriving* or *proving* the implications of a knowledge base, and it would seem at first that all that is necessary to build an intelligent system is to bring all of the knowledge together and derive the entailments. Providing mechanisms by which the system's external actions can influence the world and by which changes in the world (system generated and otherwise) can be recorded in the system's knowledge base then closes the loop. On each cycle, the system derives the appropriate action and observes any changes that have occurred internally and externally, and then goes back to deciding what to do next.

(Note that such a system follows the Knowledge Representation Hypothesis, Physical Symbol Hypothesis, and Second Order Platonism Principle, since all of the internal elements of the system, including the steps involved in derivations, are in fact interpretable. Some implementations of logic programming techniques depart slightly from the Second Order Platonism idea in that the individual steps used in the proof methods based on refutation may not correspond to how a person would normally carry out a logical derivation. However, these individual steps are still interpretable and, more importantly, the larger-grained performance of the system can often correspond closely to human cognition.)

This is, in fact, a viable architecture for some purposes, but it is not in itself a complete solution to the problem of intelligence. One problem, of course, is tractability. Trying to deal with the implications of an entire general world knowledge base is impossible, so any intelligent system will require some means of focusing on the information that is relevant at any particular point. This is particularly important when one considers how many subtle entailments are involved in any one piece of information. Somehow, we must avoid being mired in these usually trivial implications and yet have them available when comprehension depends on them (Charniak, 1982).

Another major problem, not unrelated to the first, is the problem of representation. The world, as we think of it as human beings, is a very complex, multidimensional entity. It consists of concrete as well as abstract entities that persist and change over time, that influence each other, that have various relations to each other, and that are organized into various conglomerate structures—a recursive description, since conglomerations persisting over time are themselves entities that can participate in relations with other entities. It is by no means clear how to develop representational mechanisms capable of handling all of the subtleties of our human knowledge of the world, let alone to meet the additional requirement of computational tractability. As a result, artificial intelligence has implicitly adopted a two-part strategy to address this complexity. Many projects in artificial intelligence, often the more theoretically oriented projects, are essentially aimed at isolating one particular issue in representation or reasoning, on the implicit assumptions that: (1) only through a thorough understanding of these separate issues will we understand how they work together, and (2) the approaches taken to these separate theoretical perspectives will in fact prove to be more or less coherent with each other. These sorts of projects attempt to deal with issues such as:

1. how to represent and reason about difficult dimensions of the world such as time, space, or causality;
2. how to represent and reason with mental representations of useful aspects of experience such as "cases" (memories of events) or "models" (representations of objects and systems);
3. how to reason with uncertain or incomplete information, and with uncertain knowledge;
4. how to develop coherent structured plans from smaller components;
5. how to reason about the actions, beliefs, and goals of other agents;
6. how to generate and comprehend natural language utterances; and
7. how to learn various aspects of these processes.

The other part of this global implicit research strategy consists of projects which periodically attempt to merge the current state of understanding on several issues to see how much can be done within the current state of understanding.

This section and the next two will sketch out several of the issues upon which research efforts have been focused in artificial intelligence over the last few years. This overview will not be exhaustive by any means, and it is too brief to be really representative. It is intended to provide a sense of how the very general problem of computational intelligence can be approached from a variety of more particular perspectives. In a later section, Future Directions, the nature of the problems encountered in putting some of these issues together in an integrated approach to a complex problem is considered.

Common Sense Reasoning

Definition 5
Intelligence is common sense reasoning

In sum, then, artificial intelligence would like to address the full range of activities involved in an agent's ability to pick up information from the environment, to internally store meaningful generalizations over that information, and to internally manipulate representations of the world that are faithful enough to the real world to provide a useful basis for determining how the organism ought to act. This may at first sound like a description only of relatively low level aspects of intelligence, such as physical navigation in an environment. In fact, it does describe abstract planning and reasoning as well. Several of the particular approaches taken to this general problem involve attempts to capture aspects of common sense reasoning. Although some of these research directions may appear rather esoteric, it is important to realize that they all reflect important perspectives on the general capacity of human thought, the products of which constitute the knowledge base we call a library.

Nonmonotonic Reasoning

The basic reasoning process presented above in the section on entailment has certain important limitations, deriving essentially from the fact that such classic reasoning systems cannot deal with uncertainty in any fashion. All statements in the system are either true or false. Moreover, since the inference rules are *sound,* that is, never produce an expression that is not entailed in the database, there is a sense in which logical deduction does not really add anything new to a database but, rather, only makes explicit what was already implicit. Thus, the only really new information that is available for the system must in fact come from the outside. This point is all the more important when it is realized that not only is it important *in fact* for many systems to acquire new information, for instance, as circumstances change; but it is also important *in principle* because it is impossible for any set of expressions to capture the full detail of any components of the real world. The expressions are rather an approximate description, and we will often have occasion to want to update or fine tune that description, even when it concerns static situations.

But there is an important restriction on the nature of the new information that is permitted in classic inference systems. In classic (monotonic) logic, any belief (theorem) that is derivable from a given set of data (axioms) is also derivable from any superset of that data. That is, knowing something new cannot result in the deletion of anything previously believed. This is clearly not the case in human reasoning where we tentatively hold, and of necessity act on, all kinds of beliefs, some of which turn out to be simply false, and many of which turn out to have exceptions. Similarly, it would be an impossible restriction to have any intelligent system require that all of its beliefs be certain before it takes any actions based on them. In medical diagnosis, for instance, data are often uncertain or unavailable, and the rules that draw conclusions from the data are themselves only probabilistic.

Artificial intelligence has taken a number of approaches in regard to this problem. The major approach has been to use informal methods such as certainty factors in rule-based systems and cancellation in inheritance networks. While these methods seem to work well enough to support many important theoretical and practical developments in artificial intelligence, there are some important drawbacks. For one, although many would equate their informality with the notion of common sense, they are really just syntactic devices, and if common sense is anything, it is semantic. The point is that a rule (for instance) that states that a given conclusion is warranted under certain conditions does not, in itself, tell why the conclusion is likely, and thus does not support reasoning regarding the validity of that rule (for instance, under unusual circumstances). Thus, from a theoretical point of view they leave us with an inadequate account of this very important aspect of thought. Second, this informality leads to uncertainty regarding a system's performance. Not only is the system's performance uncertain, it can be extremely difficult or impossible to even get an estimate of how likely an incorrect conclusion might be. This sort of problem will not necessarily be critical in all applications, but it certainly is when we contemplate handling applications such as nuclear power plant maintenance, space station life support, and even national defense.

The other major approach to the problems of dealing with uncertain information is to try to develop formal models of these processes. For instance, if a system is to act on uncertain beliefs, it must have a mechanism for retracting incorrect beliefs and the conclusions that have been drawn from them. It must also have a mechanism for expressing the general "default" or "typically true" beliefs that compose the major part of human general world knowledge, and for generating those uncertain inferences that such knowledge would allow, while blocking the inferences that are clearly unwarranted. Truth maintenance systems (Doyle, 1979) and their descendants are approaches to the former problem of how to eliminate no longer supported beliefs, while the general area of nonmonotonic reasoning has developed to deal with the latter (Ginsberg, 1987).

The general idea behind nonmonotonic reasoning is to capture ideas such as "in the absence of information to the contrary, a car will start when one steps on the gas and turns the key." This assumes several defaults such as that there is gas in the car, the fuel system is working, the electrical system is working, there is no anti-theft device set, the car is in the correct gear to allow the starter to be activated, etc. In a conventional logic, one would have to specify all of these conditions to the implication. However, in many cases the exception conditions may not be enumerable, or they may themselves decompose to a non-enumerable set of conditions, e.g., all the reasons why the electrical system might not work. Moreover, this sort of background information is usually not stated in the case of communication, and is very time-consuming to gather or to reason about, even in the case of single agent activities. It simply makes no sense (in most situations at least) to worry about *why* the car might not start until it actually fails to do so. What is needed is a way of warranting the default assumption that if one wants to start a car, one needs to just press on the gas and turn the key, *unless one has reason to believe that that will not work.*

In general, there are two criteria to be met by all approaches to difficult problems in computational reasoning, including nonmonotonic reasoning. First, they must accurately reflect at least part of the underlying intuitions we have regarding the sort of reasoning in question; second, they must be computationally feasible, at least within some constraints. Not surprisingly, it has proven very difficult to develop formal models that meet both of these criteria adequately. Computational models of nonmonotonic logics are very slow, and so far are thought to have (in the general case) the very unfortunate property of being *non-semidecidable.* That is, not only are they not guaranteed to produce an answer in finite time when the answer is "no," as are standard (*semidecidable*) first-order systems, but they are also not guaranteed to produce an answer in finite time in the positive case either. Not only does this mean that such reasoning systems tend to be very slow, i.e., the *worst case* complexity analysis tends to be reflected in the *average case* performance; even more important, it means that the systems cannot be guaranteed not to make mistakes. That is, if the system is checking the consistency of a conclusion it would like to draw, it may never return with the reason for which the conclusion is in fact inconsistent. As a result, the practical applications of nonmonotonic reasoning are still rather limited. But it is one of the most active areas of theoretical investigation, and many expect it to eventually produce important practical results as well as important theoretical implications for other areas such as knowledge representation. In fact, in recent proceedings such as those of AAAI and IJCAI, work in nonmonotonic reasoning is frequently categorized under knowledge representation. The following classic examples provide a sense of the approaches and issues that are being explored in this field.

- Closed-World Assumption

Conventional databases assume that any fact not contained in the database is not true. This avoids the necessity of explicitly recording all of the possible negative information that the system might otherwise need to record (e.g., all the possible flights that do not exist in a travel agent's database). When systems are capable of deriving entailments from their databases, however, this notion of the "closed-world assumption" is complicated by the fact that the system might implicitly contain facts, and one can no longer assume these to be false just because they are not explicitly stored. In other words, one may now assume to be false only that information which is neither contained in the database nor derivable from it. Thus, Reiter (1978) attempted to extend the closed-world hypothesis to the deductive database case using the idea that a closure could be calculated for a database by including the negation of any positive *ground literal*—a basic unquantified expression that does not contain any connectives ("and," "or," implication, etc.)—that is not entailed by the database. Of course, the reason for drawing these new inferences is to be able to use them for further reasoning processes. However, the complexity of the problems involved in nonmonotonic reasoning is nicely illustrated by the simple fact that this intuitively appealing idea does not necessarily produce a consistent database closure. For instance, from (P or Q) both $\neg P$ and $\neg Q$ are added to the closure since neither P nor Q is itself entailed, but all three, taken together, are not consistent. Fortunately, there are useful things that can be done with the closed-world assumption in spite of this, by restricting its application. For instance, it is possible to talk about the closed-world assumption with respect to a single predicate or set of predicates. In addition, the Horn clause formalism, upon which logic programming languages such as PROLOG are based, has the property that the closed-world assumption augmentation of a consistent database is in fact still consistent, and this is the foundation of the "negation as failure" implementation of the NOT operator in PROLOG.

- Default Logic

Reiter (1980) developed another approach to nonmonotonic reasoning in which default statements are represented as rules of inference of the following form:

$$\frac{\alpha(x) : \beta(x)}{\gamma(x)}$$

These are to be read as meaning that if $\alpha(x)$ holds, and $\beta(x)$ can be consistently assumed, then infer $\gamma(x)$. The usual situation, in which $\beta = \gamma$ (i.e., what must be consistent is simply the possible conclusion), is referred to as a *normal* default rule. As in the case of the closed-world assumption, this formalism is used to augment the initial set of beliefs, in this case by applying the default rules to the initial database. One interesting feature of this approach is that different augmentations or *extensions* are possible given the same initial database and set of default rules. For instance, given the following default rules and facts, it is possible to conclude either that Nixon is a pacifist or that he is not, depending on the order in which the default rules are applied:

$$\frac{\text{Quaker}(x) : \text{Pacifist}(x)}{\text{Pacifist}(x)} \qquad \frac{\text{Republican}(x) : \neg(\text{Pacifist}(x))}{\neg(\text{Pacifist}(x))}$$

Quaker(nixon)
Republican(nixon)

These can be considered alternative belief states that a system could arrive at based on the original knowledge state, but unfortunately, it is difficult to do much reasoning about the processes that lead to these alternate belief states because the default rules are themselves not objects of the logic language per se and cannot be themselves the objects of reasoning processes (the same problem referred to above regarding rules in general).

- Circumscription

One of the most influential approaches to nonmonotonic reasoning is that of circumscription, introduced by McCarthy (1980, 1986). The basic idea involves the development of a *circumscription formula* which is another way of augmenting a database to include information that has not been explicitly stated. The circumscription formula is designed to add to the database only information which *must* be true with regard to a particular predicate or set of predicates given the initial database. That is, it is designed to add whatever is entailed by the *minimal* model (state of the world) that is consistent with the initial database. The idea is used for default reasoning by introducing special "abnormality predicates" that express exceptions to general rules and then circumscribing them so that things are only believed to be abnormal that of necessity must be abnormal given the initial database. For instance, suppose we have a database containing the following:

$$(\forall x)\ \mathrm{Thing}(x)\ \&\ \neg \mathrm{AB1}(x) \supset \neg \mathrm{Fly}(x)$$
$$(\forall x)\ \mathrm{Bird}(x) \supset \mathrm{Thing}(x)\ \&\ \mathrm{AB1}(x)$$
$$(\forall x)\ \mathrm{Bird}(x)\ \&\ \neg \mathrm{AB2}(x) \supset \mathrm{Fly}(x)$$
$$(\forall x)\ \mathrm{Ostrich}(x) \supset \mathrm{Bird}(x)\ \&\ \mathrm{AB2}(x)$$

AB1 expresses abnormalities that are possible to the general rule that things don't fly, whereas AB2 expresses abnormalities that are possible to the general rule that birds do fly. Note that all birds are AB1, although they may not fly anyway, since some of them may be AB2 as well. Circumscription, via a step through second order calculus (i.e., one treating predicates as variables) derives the following new (first order) rules which are not directly derivable from the original database within first order logic.

$$(\forall x)\ \mathrm{Thing}(x)\ \&\ \neg \mathrm{Bird}(x) \supset \neg \mathrm{Fly}(x)$$
$$(\forall x)\ \mathrm{Bird}(x)\ \&\ \neg \mathrm{Ostrich}(x) \supset \mathrm{Fly}(x)$$

- Multiple and Default Inheritance

As the last two examples illustrate, a good deal of nonmonotonic reasoning is concerned with issues which are very similar to the kinds of inferences made in inheritance networks, and a good deal of work in nonmonotonic reasoning is explicitly aimed at developing cleaner semantics for issues such as multiple inheritance and inheritance with exceptions (Etherington, 1987a, 1987b; Horty & Thomason, 1988; Touretsky, 1986). These involve problems of inconsistency often encountered when information about a concept can be inferred from more than one superordinate concept in a knowledge base.

Reasoning from (Informal) World Models

Much of human reasoning seems to be based on (sometimes) informal manipulations of various representations that we have or construct about the world. The following sections introduce several of the approaches taken along these lines. These approaches overlap now, and will probably overlap more in the future. They involve (in varying measure): (1) the use of similar previous situations to evaluate new ones; (2) qualitative reasoning about space, time, and the ways in which physical systems behave; and (3) the development and use of (temporary) models of devices and situations.

- Case-Based Reasoning

The case-based reasoning paradigm is largely based on the notion of episodic memory, as first developed in psychology by Tulving (1972) and later developed in artificial intelligence by Schank and his students, especially Kolodner (1984). This approach begins with the observation that much of what is stored in human long-term memory consists of particular events, including their characteristics such as what actions were taken and what results occurred from these actions. Much of our current problem-solving activities can be construed as using these past examples as templates for our current actions, as opposed to abstract reasoning based on the principles that have either been directly acquired, as in school, or abstracted from experience (Kolodner, 1988; Rissland & King, 1988). Some case-based reasoning approaches involve actively analyzing the past experience to produce a new problem-solving solution, whereas other approaches involve only showing that a previous situation is similar enough to a current one that it ought to be considered *relevant* to the current situation. The latter approach has largely been limited to legal reasoning from *precedents* (Ashley & Rissland, 1988) and appears to be of less general interest than is the former. However, both ideas capture an important feature of human reasoning, and recent efforts have been made toward integrating this approach with that of reasoning from general knowledge (e.g., Rissland & Skalak, 1989).

- Qualitative Reasoning

People clearly move about in physical space and make predictions about the movements of other objects (e.g., trajectories of thrown objects or the paths of accelerating self-propelled objects) that would require tremendous computational resources if they were calculated in the full details of the laws of physics, and they do so with remarkably little effort or even conscious thought. Apparently they are utilizing a level of description of knowledge of physics which is far more general and computable (to say nothing of learnable) than is scientific physics. The objective of the field of qualitative physics is to develop a theory of this kind of knowledge in order to support common sense reasoning about movements in the world and also to serve as a sort of preliminary guide for the application of full scientific physics when that is appropriate.

The key to this approach has been the attempt to develop low-resolution abstractions of the real world that capture only those aspects of the world which are relevant to the kind of reasoning that is being performed (de Kleer & Brown, 1984; Forbus, 1984; Kuipers, 1984, 1986). For instance, in many situations all that really needs to be represented about a physical system can be captured in signs (positive, negative, zero), that indicate which way a system is going, (e.g., the sink is filling or emptying, or the ball is rising or falling), and a representation of what sorts of boundary conditions will transfer a system from one sign to another. A more detailed but still qualitative level of description is the use of inequalities (e.g., x will continue to be larger than y until event e occurs).

Unfortunately, work in this area does not seem to be leading towards a general representation and inferencing mechanism for time and space, but rather seems to be focused on the representation of, and reasoning about, particular physical systems. This does not appear to be a temporary strategic emphasis, but rather reflects a general property of the approach, which, much like model-based reasoning (see next section) is based on segmenting reality into a series of zones within which relatively simple representations hold and between which transitions occur. This dependence on reasoning about particular devices or situations provides a number of advantages such as a convenient framework upon which to base the causal structure of events and the constraint structure of the domain. But the lack of such decomposability in general time/space movements severely limits the likelihood of general success for the present approaches to naive physics as an approach to unconstrained movement (i.e., for general kinematics). On the positive side, such approaches have been developed for several complex industrial systems such as turbojet engines, power plant condensers, mechanical systems in a helicopter, and semiconductor fabrication (Cohn, 1989). Not only are these important problems in themselves, they are good examples of the sorts of domains in which it would be reasonable to hope to be able to extract executable knowledge from text in the foreseeable future. This work, like some of the work in the related area of model-based reasoning, points to the sorts of domain-specific models of reasoning that will be required for this kind of deep natural language processing.

- Model-Based Reasoning

A number of research efforts in artificial intelligence and cognitive science have focused on how people use models of particular entities in the world to reason in very specific ways about those entities. In cognitive science, work has proceeded in two related directions, both known as "mental models" research (Gentner & Stevens, 1983). Johnson-Laird (1983) has developed a theory about how people perform syllogistic reasoning and similar abstract reasoning processes in which it is assumed that they develop (often consciously) simplified, usually very concrete, models that capture part of what the problem implies (e.g., a typical situation of which the problem statement would be true). He has further proposed that people read their conclusions from these models rather than use a rule-based or logical approach, and he has found evidence that people's errors correspond closely with those that would be predicted by the use of these sorts of approximate models. In spirit, though not in detail, this approach is similar to that of naive physics, and might even be called "naive logic."

A second approach to model-based reasoning in cognitive science is very device specific. For instance, Norman (1983) has investigated the mental models that people have of a variety of calculators and explained their lack of optimal procedures for these calculators based on their mental models. This sort of approach is seen also in much work in the area of intelligent tutoring systems, in which there is often an attempt to model a particular device that is being taught and the student's knowledge of that device.

The major approach to model-based reasoning in artificial intelligence attempts to base reasoning, particularly diagnostic reasoning, directly on what is known about the physical device that is modeled (Davis, 1984; Reiter, 1987). In other words, this approach attempts to develop diagnostic "reasoning from first principles" as an alternative to the traditional "compiled-knowledge" or rule-based approach. The approach requires the ability to specify a well-defined model of the device, including all the constraints that exist between the well-defined components of the device. As a result, this approach has so far been restricted to electronic, hydraulic, or similar devices that have such well-defined internal structure and behavior. (Some aspects of medicine are at the fringes of the feasibility of this approach.)

The diagnostic reasoning process begins by comparing the observed behavior of the real device (e.g., the input and output readings of an electronic device) with that predicted by the model. It then proceeds to generate hypotheses which essentially consist of the lifting of constraints regarding different parts of the model. In addition to hypothesis generation, the process considers issues related to reducing the search space of possible problems, and of discriminating between alternative explanations.

In comparison to other approaches to diagnostic reasoning, model-based reasoning is relatively device independent. That is, the new information required to diagnose a new device is essentially a declarative description of the device itself, rather than a detailed procedure that relates to that device. As such, it is far more feasible to imagine the direct extraction of text-based information into such a system than it is in the case of rule-based systems. In essence, the device-specific knowledge required for model-based reasoning is of the sort which is more or less directly stated in textual documents, whereas the procedural knowledge required in rule-based systems is usually unstated. Of course, as presently developed, this form of case-based reasoning is somewhat limited in the scope of applicability.

Reasoning about Time and Action

Many of the reasoning tasks undertaken by artificial intelligence involve the dimension of time and sequences of actions that occur over time. These include obvious areas such as planning and prediction, and also areas such as natural language understanding, explanation, diagnosis, learning, problem solving, and spatial reasoning, all of which deal with the temporal location of events. Unfortunately, at least from the perspective of developing a general theory of temporal reasoning, each of these areas (and others as well) has developed a different approach to dealing with time, often by trying to deal with it as implicitly as possible. For instance, a basic state space approach to problem solving consisting of a set of possible world conditions (or "states") and a set of permissible operators that change one possible state into another, simply assumes that the order of operator applications (as reflected in the shape of the search tree developed) indicates the temporal order of the operators, and that that is all that is relevant regarding time.

This section deals with research areas that need to be relatively explicit about time. These research directions began under the concept of "planning," which has long been considered one of the major general topic areas in artificial intelligence. In recent years, planning has divided into a number of related research topics.

Planning

Planning involves the selection of a series of actions designed to achieve a goal. In a sense, this is nearly synonymous with all of cognition, and is clearly an extremely complex reasoning process. General planning involves knowledge about: temporal relations; physical space; causal relations between actions and states; changes in physical and social conditions; uncertainty regarding the nature of the outcomes of particular actions (and even the ability to carry out particular actions in the world); the beliefs, plans, goals, and intentions of agents; planning knowledge (meta-knowledge); and strategies. It also involves the need to obtain information dynamically and the ability to deal with conflicts and interactions among goals.

As a result of this complexity, early work in planning, particularly work attempting to develop formal approaches to the planning process, was based on several severe simplifying assumptions regarding the nature of the world.

The prototypical example of the finessing of the question of time by using such simplifying assumptions is the situation calculus of McCarthy and Hayes (1969) which was the basis of much of the classic literature on planning (e.g., Fikes & Nilsson, 1971). This formalism (which is very similar to the state space search approach to problem solving mentioned above) represents the world as a series of static states and the instantaneous changes that can occur to transform one state into another. The primary advantage of this approach is that it is simple. The disadvantages are many and include the following:

1. No representation of gradual, ongoing, or delayed effects of actions.
2. No concurrent or overlapping actions.
3. No ways to deal with the various common-sense implications of actions such as the default consequences of actions when things are normal and the assumptions regarding what aspects of the world have not changed.
4. No explicit representation of time per se at all. Moreover, this approach to temporal change assumes that only one agent is causing change, and that the knowledge of what occurs as a result of the agent's actions is certain.

Even in this relatively trivial world, interesting issues were explored, such as:

- How to represent those aspects of the world that *have not* changed after an action has taken place (the "Frame Problem").
- How to monitor and repair actions that conflict or otherwise produce an unsatisfactory plan.
- How actions can be transformed by the knowledge of what conditions need to hold after they take place.
- How plans can be hierarchically produced by starting with the most general considerations and/or those considerations which are least alterable.

Moreover, some of these issues began to involve treating time in a more explicit fashion. For instance, problem decomposition approaches to problem solving and planning ran into the somewhat temporally flavored problem of subgoal interaction, one form of which is having the antecedents for an operation prematurely undone by another operation. This led to devices such as "protection intervals" during which the planner could not alter the conditions brought about by operators.

In addition, this early work began to map out a space of possible planning problems, according to such issues as whether the universe is predictable in terms of the outcomes of actions and what other agents exist and how predictable they are. As work progressed in planning, it became clear, as in other major areas of artificial intelligence, that there were too many issues to be dealt with simultaneously, and the field separated into a number of (still overlapping) specialties, each of which can be thought of as relaxing one or more of the original simplifying assumptions of traditional planning.

Temporal Reasoning

One of these specialties, temporal reasoning, is an attempt to clearly represent time as an explicit and universal dimension of knowledge representation (e.g., Allen, 1984; McDermott, 1982). This is not easy, however, since there is no simple underlying scale to use. A literal timeline, in some universal time, for instance, is not always feasible since we often do not know the exact time of events. Relative temporal information is also often problematic since it frequently leaves the relative positions of nonadjacent events unspecified.

On a more technical level, the decision to represent time as an explicit dimension leaves many representational issues open (Shoham, 1987). These include:

1. whether the basic units of time should be points or intervals, and which of these units should be the elements over which assertions should be asserted;
2. how the truth values of intervals constrain overlapping or contained intervals (e.g., if we say that *A* is true from time1 to time2, do we really mean that it is true over all intervening intervals);
3. the mathematical structure of time (e.g., whether it is bounded and whether it is a continuous scale); and
4. how exactly to formalize temporal expressions (e.g., whether time is simply to be treated as additional arguments to standard predicates or as a qualitatively distinct dimension on which atemporal propositions are located and evaluated).

Moreover, the attempt to reason about temporally situated events leads back to the issues introduced under nonmonotonic reasoning, in what are even more complex forms. For instance, Shoham and McDermott (1988) discuss what they call the ***qualification problem*** and the ***problem of extended prediction***. The former is the problem of permitting defeasible inferences (those that might turn out to not be correct) without reasoning about all of the qualifications that might limit their validity (see the section on nonmonotonic reasoning), while the latter concerns the length of time over which predictions of the future are valid, and the potential need to decompose a scenario into an unmanageable number of substeps in order to reason about the consequences of actions. In effect, the latter point is that in reasoning under uncertainty, that is, with less than complete knowledge, there is a trade-off between the reliability of inferences, which is greater with small steps, and the efficiency of reasoning, which is greater with large steps. Since there is no way to avoid the need to reason with incomplete knowledge, this seems to point to a very general dimension concerning intelligent system design.

Opportunistic Planning and Embedded or Reactive Systems

A third general area of research involves developing planning systems that are capable of dealing with the opportunities and contingencies presented by real-world events (as opposed to attempting to formulate complete plans prior to execution, as in traditional planning research). Research in this tradition tries to develop useful plan segments that respond appropriately to situations presented by the world, and is concerned with issues such as reasonable response times, planning based on incomplete information, the need to develop plans to gather new information, and the need to elaborate plans and to react to unanticipated events during plan execution (Hayes-Roth & Hayes-Roth, 1979; Georgeff & Ingrand, 1989).

Multi-Agent Planning

A large number of difficult problems are introduced into the planning process when one begins to deal explicitly with the interaction of multiple agents, but these approaches also have the power to deal with a variety of situations that go beyond those of traditional planning systems. Among the issues that such approaches deal with are the need to negotiate between competing agents (Sycara, 1988), and the coordination of distributed cooperating agents (Durfee, 1988). A good deal of this research (and of the related work on plan understanding) focuses on the nature of goal relationships between agents, and much of the work on multi-agent planning and goal relationships has focused on architectures (especially the Blackboard architecture) designed to support this sort of reasoning (Lesser et al., 1989).

LANGUAGE, COMMUNICATION, AND COOPERATION

Clearly, natural language understanding is the single area in artificial intelligence that has the greatest potential for impact on library science. It is just as important for other application areas within artificial intelligence, since we are often concerned with manipulating the products of human intelligence which are usually expressed in natural language and with interacting with human users in natural language. For instance, we are concerned in artificial intelligence with natural language interfaces to database systems, to intelligent tutoring systems, and to expert systems, and, as mentioned above, we would like to be able to augment the knowledge acquisition process by extracting knowledge directly from text.

However, natural language understanding is also in many ways the broadest and most complex issue within artificial intelligence, as evidenced by the fact that it is the only subarea within artificial intelligence that has a totally independent discipline that is concurrently devoted to the problem. This discipline, computational linguistics, has its own journal, annual meetings, workshops, etc., as well as departments and departmental subsections. (The only other independent discipline with such close ties to artificial intelligence is cognitive science, which cuts across many, even most, of the interests of artificial intelligence, including language understanding.) Two general issues, which are critical for full natural language understanding, place language understanding in the context of the sorts of issues that have been discussed above. The first issue concerns the reliance of natural language understanding on general world understanding, including the understanding of other agents.

Language Understanding is Understanding

Definition 6

Intelligence is Culture, Communication, and Cooperation

One of the most important aspects of what is meant by intelligence is the ability to communicate between, and coordinate among, agents. Clearly, much of what we communicate about refers to various aspects of the world—physical properties, time, causality, movements, etc.—and the ability to interpret these sorts of communications depends exactly on the ability to reason about these physical qualities of the world (e.g., Talmy, 1988). On one level, language is simply a refined and complex way of sharing our understanding of the physical world; on another level, language clearly enables us to produce a deeper understanding of the world than would otherwise be possible.

Plan-Based Understanding

The most significant part of the world that people communicate about is that of other people, their plans and motivations, and full understanding of linguistic communication requires the understanding of why intelligent agents took the actions (both physical and linguistic) that they did. This understanding depends on the comprehension of how goals are accomplished in the world, how they interfere and interact with each other, how beliefs and intentions generate goals, and how goals are communicated and coordinated among agents (e.g., Charniak, 1988; Kass, 1989; Wilensky, 1983). (It also depends on some implicit knowledge that human speakers share regarding how they will plan speech acts in order to assist the hearer in deciphering their intended meanings.) This understanding, in turn, is the basis for understanding issues concerning how people interact in such situations as cooperative problem solving, competition, negotiation, etc.

Explanation

These issues can all be construed as examples of explanation-based understanding. For instance, when a person says "Watch out," understanding the remark involves understanding, i.e., explaining to oneself, the reason that the remark was made, whether it refers to a physically threatening situation, the plans or motivations of another person, or both. Thus the problem of explanation can be seen as a common thread which runs through a number of issues in natural language understanding, planning, and human computer interaction.

In the area of expert systems it is critical since, at a minimum, a person must understand the reasoning process in order to be able to accept (or reject) the machine's conclusions. This is particularly important in light of the uncertain knowledge and data with which expert systems deal, and the consequent heuristic nature of their reasoning, and the very important consequences in many cases of the recommendations that they make. However, the state of the art of explanation facilities in expert systems has not caught up with the theoretical work in explanation. Most of the current explanation systems that are in widespread use are descendants of the original MYCIN technique of using rule traces to answer *how* and *why* questions (Shortliffe, 1976).

Although a rule trace is an interpretable structure, it is often not the sort of thing that a person would provide another person to explain his or her reasoning, and a good deal of work has been going on to try to generalize and improve the rule trace approach. Some of this work has approached the problem of trying to explain the strategic level of a system's behavior by using explicit strategic problem-solving knowledge (e.g., Hasling et al., 1984) while other work tries to improve the rule trace by pruning it according to issues such as the importance of steps (Wallis & Shortliffe, 1982) or the needs of particular users (Moore & Swartout, 1988). Ultimately, however, it is necessary to view the generation of an explanation as a distinct cognitive act rather than as a (possibly edited) readout of another cognitive act. Wick and Thompson (1989) have taken such an approach to reconstructive explanation by viewing it as a problem-solving activity in its own right. One important aspect of the overall problem of explanation is nicely illustrated in the case of interactive intelligent systems, in which a user can become part of the problem-solving process. In these cases, in order for the user to be able to realize the consequences of taking certain decisions she or he must have available a wide range of qualitatively different sorts of explanations.

Context Recognition and Knowledge Retrieval

A second basic issue in general natural processing is how to bring to bear the relevant sources of information necessary to interpret given inputs. For instance, a sentence such as *As the boy walked down the aisle he took a can of tuna fish from the shelf and put it in his basket* seems to invoke a description (e.g., *frame*) of a supermarket in which to understand the event. But the connections between the concepts of this sentence and the concept of supermarket are all rather weak, and on close consideration there would probably be dozens, if not hundreds, of other concepts which would be just as likely to be invoked. The problem is that such background information must be invoked—it is part of what we mean by understanding—but there are too many potentially related concepts to just generally invoke all related ideas. This problem can be construed as the question of how to bring together, in a coherent fashion, the explicit knowledge-based aspects of language understanding and the looser sorts of reasoning which seem to be involved in suggesting contexts and explanations. Suggestions along these lines differ in how much they favor explicit knowledge-based approaches and how much they favor loose probabilistic connections (e.g., Charniak, 1988; Norvig, 1989).

LEARNING

Definition 7
Intelligence is the Ability to Learn

The Western intellectual tradition has a long history of turning to learning as the key to all of intelligence. The British empiricist philosophers of the seventeenth through nineteenth centuries, as well as the behavioral psychologists of the first half of the twentieth century, essentially avoided any difficult questions regarding the nature of knowledge and the processes that utilize knowledge by attempting to uncover the mysteries of the processes by which that knowledge is acquired. Today, most cognitive scientists agree that understanding the products of learning is the key to unraveling the learning process itself. Nevertheless, learning is a critical issue from a number of theoretical and applied perspectives.

In addition to being an important theoretical issue in and of itself, it is also important as a test of the other theoretical constructs that are developed. All of the representational and inferential approaches used by artificial intelligence should be able to stand the test of learnability. In addition, learning is a critical theoretical component of other cognitive facilities, most obviously that of language. Third, from an applied perspective, the overwhelming size of general world knowledge upon which full natural language understanding and many aspects of common-sense reasoning depend seems to demand learning facilities since handcrafting the entire knowledge base appears to be unfeasible. (An attempt to build a knowledge base containing all of general "consensual" knowledge is the CYC Project [Lenat & Guha, 1990]. It is not known whether this 200 person-year project will succeed; in any case, its purpose is to build a large enough knowledge base to support machine learning of specialized knowledge.) Fourth, learning is critical to overcoming several other important problems in applied artificial intelligence. For instance, in the general area of expert systems, we are faced with the well-known knowledge acquisition bottleneck. A good deal of effort is being put into developing software to aid in knowledge acquisition, but that is not enough. We would also like to employ learning procedures to help fine tune the knowledge base of an expert system from its experience in the domain. Moreover, in the case of systems that deal significantly with the knowledge systems and goals of other agents (e.g., opponents, students, or interactive users), we would like the system to be able to induce models of those agents and hypotheses regarding their actions (e.g., reasons for the errors that students make). As mentioned above, this sense of learning, that is, the determination of a model for another agent's actions, turns out to be critical for natural language understanding as well. Perhaps the most ambitious long-term learning goal is that of developing systems capable of acquiring significant amounts of knowledge from general unconstrained text which, of course, would be a major advance on the problem of knowledge acquisition.

The general learning problem is, however, extremely difficult. It is never obvious what has to be learned (out of the general array of information present), and it is often necessary to have a good deal of background information in order to assimilate something new. In the case of trying to learn about the consequences of actions there is the particular problem of determining which, out of all of the actions which took place, was responsible for the outcome that occurred (the credit assignment problem).

In other words, learning, like the general problem of cognition and many of the other subproblems within it such as planning, turns out to be a multidimensional problem. There is no one learning problem or learning scenario in cognition or artificial intelligence, and no single learning algorithm or general approach to learning has proven capable of handling all of the situations in which learning occurs. Learning systems have approached the general problem by taking very specific positions on a number of dimensions of the learning situation.

These approaches can be categorized in several ways. For instance, one can distinguish systems based on what they are designed to learn, e.g., individual concepts, memory organization, or procedural knowledge such as problem-solving ability or strategy. They can also be classified in terms of the knowledge representation system used, e.g., predicate calculus, frames, semantic nets, memory organization packets, production rules, or plans. Probably the most common classification of learning programs involves the kind of learning situation they are designed to deal with. These include rote learning, direct instruction, learning from advice, and learning from examples. A number of factors differentiate the learning environment further, as the following questions illustrate:

- Is there a teacher or just unmediated experience?
- If there is an explicit or implicit teacher, is there a special order to the examples that are chosen for presentation to the learner?
- Do the examples include negative examples?
- Does the training set include noise (i.e., examples which are incorrectly labeled or classified)?
- Can the learning system itself generate or request test cases?
- What is the nature of the feedback regarding the presented examples (yes/no vs. why vs. results of actions)?
- Are the examples presented incrementally or simultaneously?

Most of the distinctions just described are most relevant to an approach to machine learning that assumes that the basic problem is that of inducing concepts from examples. By far the greatest amount of research in machine learning has taken this example-based or similarity-based approach to concept formation. These systems have been based on several ways in which formal expressions can be generalized: replacement of constants by variables, replacement of constants by more general elements (e.g., from a type hierarchy), or alterations of representation forms (e.g., eliminating links in a network or predicates from an expression). Many useful ideas have come out of this work, for instance, the classic work of Winston (1975) regarding the usefulness of near-miss examples in clarifying concepts and the version space search notion (Mitchell, 1977) which uses a candidate elimination algorithm to efficiently converge on a candidate concept. But many of the most promising ideas in machine learning have little to do with the traditional similarity-based learning notion of the slow empirical induction of concepts from large numbers of examples, and are rather based on a number of interesting insights regarding situations in which learning takes place.

Learning by Analogy. We clearly learn a great deal by assimilating new experiences to old ones, and mapping what was true of the old onto the corresponding parts of the new (Gentner, 1983; Greiner, 1988). Recent work has explored how the processes of analogical understanding can be employed in model-based reasoning (Falkenhainer et al., 1989).

Failure Driven Learning. One of the best opportunities to learn is when predictions are not borne out. This leads to the question of "why" and the consequent development of an explanation. The memory organization system developed by Schank (1982), Kolodner, and others (e.g., Kolodner, 1984) is based on the development of a generalization/discrimination hierarchy of episodic knowledge where the differences between events and their closest available generalizations provide the indexing terms.

Active Discovery Learning. Much creative learning involves the manipulation of known conceptual structures and the recognition of when a resulting new "idea" is worth pursuing. Lenat (1982, 1983) has developed systems capable of pursuing this sort of learning in areas such as number theory. (These systems do not prove their conjectures. Instead, they use heuristics to judge how interesting or promising they are, and they continue to explore the new structures that are proposed according to these judgments.) Recently, Kulkarni and Simon (1988) have employed historical analysis of a specific scientific discovery (the urea cycle in biochemistry by Hans Krebs in 1932) to derive and model the set of heuristics that seem to have played a role in that process.

Learning by Problem Solving. Most kinds of problem-solving activity result in new knowledge that is available if a similar problem is encountered. The SOAR system (Laird et al., 1987) captures this important observation. It is essentially an extension of the production system architecture, with two major modifications. When more than one rule instantiation is eligible, rather than resorting to a uniform syntactic conflict resolution strategy, it treats the situation as a full problem-solving situation. Secondly, it develops a new production rule to capture the knowledge that results from this problem-solving activity. This is probably the best example to date in artificial intelligence of the integration of a very general learning strategy into a general cognitive system (but see also Anderson, 1983). That is, SOAR is primarily a general purpose cognitive architecture capable of simulating cognitive activities as general as those that can be simulated by a production system, but it is capable of dropping into the learning mode exactly when the ongoing cognitive processes run into trouble. Moreover, it also reflects human learning in the sense that it learns best when it has a knowledge base that is closely related to the new information.

Explanation-Based Learning. In contrast to the slow, incremental generalization of examples that is the hallmark of similarity-based learning, explanation-based learning attempts to model the learning which occurs when a single event (e.g., the presentation of a novel example of a concept) is compared to a rich, domain-specific pre-existing body of knowledge that pertains to that event. This area overlaps with several of the others, in that the explanation processes and the ways in which background knowledge is used may be similar to analogy learning, failure driven learning, or learning by problem solving. The key point is that the system is actively trying to make sense of what is presented to it, and learns directly through that comprehension process (DeJong & Mooney, 1986; Krawchuk & Witten, 1989; Lewis, 1988).

A Note on Connectionism

A review of machine learning approaches would clearly be incomplete without a discussion of connectionism, a new, nonsymbolic computing method based on passing weights between the nodes in a nonrepresentational network which is supposed to be roughly analogous to the physiology of the brain (e.g., Schneider, 1987). A cautionary note in this regard: This paradigm, which has been suggested as a complete paradigm for all of cognitive activity, has no answers for several of the demonstrably essential properties of cognition which were discussed above in introducing the symbolic representation paradigm. In fact, on close analysis, every program of this type amounts to no more than a pattern-matching function between two finite sets. Pattern matching is a difficult problem, and having self-learning approaches to this issue is certainly very valuable, but it is certainly not all of cognition. In terms of the analysis of learning situations just presented, connectionism is essentially a new way of doing similarity-based learning. Not only does it not deal with the more knowledge-based forms of learning, it is, in certain regards, rather brittle when it comes to extending its knowledge outside of the boundaries of the original learning situation, even within a similarity-based approach.

Two other related caveats are worth mentioning here. The hallmark of connectionism is that it is nonrepresentational, that is, nonsymbolic. This has the appeal of seeming to avoid all of the hard work that dealing with knowledge representation involves. But the lack of interpretability means that in an ultimate sense a connectionist model is not an explanation of whatever it does, even if it performs well, since it cannot be explained, at least in terms of the level of analysis in which we are interested. (That is, one can explain how individual components are working, and how the overall system works as a system, but one cannot talk, in general, in terms of the objects of the world and objects of thought that constitute the universe of discourse regarding cognition.) Systems without interpretable components present first, an impossible software engineering task from the human engineering point of view, and second (aside from the human comprehension issue), a fundamentally impossible system design task because the various interactions between components of the system have no basis for coherence. That is, assuming that all of cognition cannot be modeled as one overall mapping from the set of inputs to the set of outputs—the arguments against this possibility are too numerous and overwhelming to mention—any attempt to simulate general cognition in a connectionist framework must assume a large number of interacting connectionist modules. (Even within very circumscribed problem areas, connectionist models are constrained to sample densely from the set of input possibilities and cannot generalize very far outside the realm of that sampling.) For the outputs of one module to make sense in the context of any of the other modules to which it might be sent, it would appear necessary for it to have a meaningful interpretation in the world. There appears to be no other basis for the internal coherence of a system (Metzler, 1990).

FUTURE DIRECTIONS

Bringing It All Back Together

The previous sections have outlined several of the important issues and perspectives that are currently being pursued in artificial intelligence. But, as mentioned above, there is something of a dialectic, or rather several dialectics, in artificial intelligence between the fine-grained perspectives and the more general. In a general sense, this is just a dialectic between levels of granularity: fine-grained analysis vs. coarser-grained synthesis. This general dialectic is also flavored by related (but not equivalent) contrasts between issue-oriented vs. architecture/system-oriented research and between theoretical vs. applied research. This section will relate these issues to the state of the art of artificial intelligence, and more particularly, to the state of the art of expert systems.

Current Expert Systems

The current state of the art of expert systems (and this is generally true of other artificial intelligence applications as well) is that they are capable of carrying out very complex tasks, and doing so in ways that roughly correspond to how people would carry out the task, but only for tasks that are, in a sense, relatively homogeneous. That is, there are no systems that are capable of dealing with even a large number of the theoretical issues or dimensions described above, in a single coherent, integrated way. And this is probably the greatest challenge for the field today. While there are plausible models of many interesting and important aspects of thinking, we do not have a plausible model of how all of these aspects can be manifestations of a single theory of knowledge representation and reasoning. And yet, even from an applied point of view, it can easily be shown that any one of these forms of reasoning can be important or necessary for certain tasks. Such a unified theory of knowledge is critical if we are to achieve the long-term goals of developing knowledge bases capable of supporting a variety of tasks and of extracting significant parts of these knowledge bases from natural language documents.

This section will (1) briefly point out a set of desirable characteristics for expert systems, (2) provide an example of two systems in one domain that illustrates the contrast between traditional homogeneous task-based systems and those attempting to deal in more flexible ways with a variety of knowledge-based tasks, and (3) illustrate how the development of such flexible expert systems is dependent on progress on the theoretical issues discussed above.

Design Requirements for Intelligent Interactive Systems

Metzler (1989) characterizes the present generation of expert systems as "task-based" systems, which lack the flexibility necessary to utilize a knowledge base for a variety of expert tasks or to interact in a number of important ways with a user. A number of design requirements must be met in order to develop expert systems that can support the flexible human computer problem-solving interaction necessary to address more typical real-world problems. Some of these requirements follow.

Ability to Address Multiple Problems

Most real problems of expertise involve a number of separate tasks. For instance, design and diagnosis are frequently two sides of a single domain of expertise, which might also include "redesign" or design modification. Typically, the same person or persons would do all of these tasks, and would do them all on a particular occasion for a particular client. A system to perform these tasks or to aid a human expert should ideally be capable of reasoning flexibly in any of these modes and also capable of bringing these different modes to bear on a single case. Note that this is a stronger requirement than the often-expressed goal of expert system development: the ability to use a single knowledge base as the foundation for a number of single task-oriented expert systems, even if the tasks are not integrated into a single system.

Integration of Multiple Experts and Approaches

The current state of the art of expert system development involves not only very specific tasks, but also a single coherent approach toward that task. There are at least two major aspects to this issue. The first involves the point that human experts often take different approaches to solving the same problem. In design, for instance, it is possible to begin in a top-down way by looking at the overall situation, or in a bottom-up way, by accumulating information about all the details of a particular situation. It is extremely difficult to try to incorporate widely different strategies in a single coherent problem-solving strategy, but, in fact, this seems to be exactly what is needed to capture the flexibility of even a single expert. Part of the inflexibility or lack of common sense of current systems is probably attributable to this lack of ability to take alternative approaches to a problem, and especially to reason about why a particular approach would be a good idea in a particular situation. Current expert systems lack the robustness inherent in having a team of experts work on a problem together. In fact, part of the current accepted practice in expert system development is to avoid trying to accumulate system knowledge from more than one human.

Multiple Relevant Dimensions

The second point regarding the integration of multiple approaches involves the point that many real-world problems involve many qualitatively distinct factors. For instance, design problems often involve trade-offs on dimensions such as cost, reliability, various measures of performance, ease of maintenance, ease of manufacture, etc. Strategic decision making often involves trade-offs on dimensions such as costs, potential gain, potential risk, etc. These trade-offs involve the ability to compare qualitatively different sorts of considerations.

In the human case, teams of experts provide an additional problem-solving advantage when each member can contribute specialized expertise that would be difficult to find in a single person. In these situations, it is not uncommon to have the experts working on individual aspects or dimensions of a problem act as advocates for the importance of the considerations they have been dealing with, and for the recommendations to which their part of the whole problem has led them.

Conflicting Information, Advice, and Requirements

One key to the development of systems capable of dealing with multiple criteria within a problem space as well as the integration of the advice of multiple knowledge sources, is dealing with the conflicting conclusions that such knowledge sources will provide. The key to dealing with these conflicts seems to be developing a richer knowledge base that explicitly includes the reasons for and consequences of various actions, so that potential actions can be reasoned about, not just invoked.

Interactive Control

A user needs to be able to redirect the problem-solving activity in a flexible manner, e.g., to return to previous points in the problem-solving process, or to impose decisions or criteria.

Explanation

The key design requirement placed on the system by the need for interactive, shared control is a full explanation facility that enables the system to explain to the user why actions were taken, what the alternatives were, and what the consequences of hypothetical actions might be. These facilities and the data structures which they require ought to be the basis also for communication among the semi-independent problem-solving modules.

Learning

A true expert system ought to improve its performance as a result of experience in the domain.

An Example: ELAND vs. ISLAND

The contrast between the general state of the art of expert systems and the desired goal of the field is nicely illustrated in the comparison of ELAND (Tanca & Ceri, 1986) and the AT&T-supported research on ISLAND (Metzler & Williams, 1988). Not all current systems are as simple as ELAND, but the majority are. Moreover, even quite complicated systems tend to be rather homogeneous in their reasoning processes. Both of these systems are intended to aid in the process of local area network design, but there the similarity ends. ELAND's task is to recommend the general characteristics that a particular implementation should employ, such as the type of network topology (ring, star, or bus), based on the general characteristics of a particular situation. It does not deal with the details of a particular situation, such as where the nodes are physically situated, distances between them, physical barriers to traverse, etc. ISLAND's goal is to simulate the full range of expertise that would be provided by a human expert in this field, including diagnosis of problems, designing a network and modifying a network, and it is intended to deal with the full reality of a particular implementation, not just general characteristics of it. (ISLAND is still under development, so while many of the design criteria have been partially met, none have been fully satisfied.) Moreover, the design and redesign problems address a large number of qualitatively separate issues such as topology, software and hardware choices, costs, reliability, performance, security, capacity, and extendibility. These issues do not reduce to some single underlying dimension. There is no simple overall "goodness heuristic" that permits all of these considerations to be thrown into a single search space. Rather, the issues must be dealt with explicitly, and they must often be explained to the user/client who must make an informed decision regarding the possible trade-offs. The design problems are further complicated by the need to deal with constraints such as using a particular vendor's products; using equipment that is already in place; and using connections to other networks, mainframes, databases, and special peripheral devices.

The ISLAND Architecture

This section briefly outlines the architecture that we have been developing to provide the kinds of capabilities that were identified in the previous section. It is intended to illustrate how different such an architecture must be from that of a typical rule-based expert system. The overall metaphor for the design of ISLAND is that of a cooperating group of experts who are under the guidance of the user/client. The notion of cooperation among expert modules and between the modules and the user places a great emphasis on the problems of communication and explanation. The basic components follow:

1. *Experts and the Interface.* The experts are semi-independent systems that each embody the expertise relevant to a particular issue in local area network (LAN) design, such as topology, medium selection, budget, security, communications (i.e., bridges, gateways, and protocol compatibility), client requirements, software and hardware selection, servers, installation and maintenance, and traffic load. Each expert reasons about its domain, posts the results of its reasoning in terms that are meaningful for the other experts, and interprets the relevant results of other experts. One special expert is the user interface expert which translates the user's control instructions into instructions that the system experts can carry out. This expert is also responsible for gathering information from the user about a particular design or diagnosis problem.
2. *Knowledge Base and the LAN Design.* The knowledge base is a frame/inheritance-based representation of all the objects and concepts that are relevant to this domain. At the more general levels are generic concepts such as computer, storage device, protocol, etc., while at the more specific levels, details of specific products are recorded. The representations of concepts include constraints on how they may be used (e.g., connected together) and information concerning various aspects of their performance. The particular design that is being developed or analyzed by the system is part of the knowledge base. It is distinct in that all parts of the design are particular "instances" of general concepts rather than concepts per se, and also in that the system treats such structures as temporary data structures rather than as part of the permanent knowledge base.

3. *Annotations and Dependency Pointers.* The most difficult aspect of this knowledge base involves structures to represent the abstract nature of the planning process itself; in particular, the existence of conflicts and the reasons for which design decisions were taken. These form the basis for explanations to the user, and for explicit problem solving, especially conflict resolution, by the system.

   Our first approach to this issue involved adding a structure to each piece of the LAN design structure that recorded the important reasons for which it was created, i.e., which expert module made the decision, when it was taken, what the relevant considerations were, what alternatives were considered and rejected, and (when possible) why the alternatives were rejected. The basic difficulty here is determining what information is worth preserving, since it is impossible to record everything. We are currently exploring a more dynamic approach to recording this information which is essentially a variant on the idea of a Truth Maintenance System (Doyle, 1979). In this approach, pointers are set from each part of the LAN design to other parts of the LAN design, including problem-solving abstractions such as *requirements* and *conflicts*, to record why the LAN design component was decided upon. The pointers are themselves annotated to record the nature of the reason that they record. Question answering in this mechanism requires the ability to trace the pointers that are relevant to a particular type of question. Reasoning about conflicting requirements can also take advantage of such pointer structures.
4. *Control Requirements and Control Mechanisms.* The requirement for flexible interaction between system and user necessitates the ability to redirect the system's current activity by resetting goals or priorities, and the ability to return to past problem-solving states (while perhaps retaining the current state for future reference). We are attempting to capture this flexibility by basing the control architecture around two data structures known as *concerns* and *calls.* The concerns are user-oriented concepts that capture what a user is trying to do at any particular point, while the calls are expert module-oriented, and are specific tasks that the experts know how to carry out. The concerns are decomposed into the more atomic calls by the interface module, and the system maintains agendas of both concerns and calls for the system to act on. An attempt is made to keep the calls and concerns fairly specific so that the user is returned to at fairly tight problem-solving intervals and is thus kept closely "in the loop."
5. *Explanation Types and Mechanisms.* The key to cooperative problem solving between user and system and between the modules of a system is the communication of appropriate information. We have been developing mechanisms to communicate the following types of explanations, which we have identified as critical to enabling the

user to direct the system's problem-solving activities, to ensuring that the user is taking responsibility for what the system is reasoning about, and to making sure that users learn from the experience of using an intelligent workbench such as ISLAND.

Domain Knowledge. Although this amounts essentially to querying the knowledge base of concepts, there are difficult problems to be addressed, such as comparisons between different objects and questions regarding the scope of the system's knowledge.

System (Procedural) Knowledge. The first pass at explaining the system's knowledge of actions to take involves reading out system rules in an intelligible format. Some additional help in this regard can be obtained from system knowledge of entities such as priorities. Higher level explanations involve explaining the system's strategies. For now, we are attempting to address this issue in terms of using, and reporting to the user, system goals that are as explicit about the reasons for actions as we can make them.

Reasons for Current and Past Actions. This is done as in a standard production system by inspecting the way that the current or past production rule matched the contents of working memory.

Reasons for Actions that are Not Taken. The reasons for which a current possible action is not being taken by the system can be understood by looking at rules that would take the action in question and seeing why their conditions are not matched or why some other action is given preference. The question of earlier possible actions that were not taken is more difficult, but is in practice a very common situation. We are exploring approaches to this problem based on the knowledge-base partitioning method known as a *context mechanism.* This method allows the system to incrementally generate environments in which the results of all previous actions are visible but from which future actions will remain hidden when they occur. At a future time, the system can return from its current context (from which all intervening actions and their results would be visible) to the earlier context which hides the intervening actions and results, and look at a question as if the earlier context were the present state of the system.

Hypothetical Situations. Questions regarding hypothetical actions can similarly be addressed by creating temporary contexts and adding the "what if" information to them.

Questions Concerning User System Interaction. Our general goal is to make the system's actions as transparent to the user as possible in order to support the most flexible possible interaction. Our general strategy to try to achieve that goal is to design the system around objects such as concerns, calls, and system actions that are as close as possible to how humans think about this problem domain. The test of using these objects to serve as explanations is at once a critical way of determining how cognitively motivated these objects are, and a way of providing exactly what a user needs to drive such a system.

What Do We Need for the Next Generation of Expert Systems?

The ISLAND project illustrates the need for better fundamental understanding of virtually all of the particular theoretical issues discussed above. For instance, it is a horribly nonmonotonic problem. Not only is it *possible* for a new fact to invalidate previously determined information such as a previously determined design decision, but, in fact, there are several concurrent qualitatively distinct lines of reasoning taking place each of which, in general, can be *expected* to invalidate parts of the reasoning processes already carried out by the others. The objects of this domain call, in many cases, for complex knowledge representation techniques, including examples of multiple inheritance as when an object is both a computer and a fileserver. The domain includes various aspects of temporal and spatial reasoning not currently being explored, some of which might best be based on explicit reasoning from cases or from models. The design process is very similar to that of planning, particularly in regard to subgoal interaction, and clearly requires "opportunistic" as well as retractable strategies. The need to communicate with a user and between modules requires a form of plan-based communication and understanding, and a form of distributed cooperative problem solving. We would certainly like the system to benefit from its experience, perhaps by storing case histories of partial and complete plans, and perhaps even by storing planning histories that capture examples of the reasoning process itself, rather than just the products of the process. Such learning would clearly have some of the qualities of explanation-based learning, and would ideally also involve the ability to learn procedural information. Many or most of the complex real-world problems that are most in need of artificial intelligence applications quickly broaden out into this sort of open-ended cognitive landscape. What keeps this problem from being totally intractable is that the domain, as heterogeneous and complex as it is, is still relatively well defined.

CONCLUSION

The State of Artificial Intelligence

The clearest implication of what has been said up to now is that, contrary to what some critics have claimed, artificial intelligence is in a very healthy state as a science and discipline. It currently consists of three overlapping general sorts of activities which mutually inform and enrich each other. One consists of a very large body of practitioners using a set of relatively well-understood tools that are capable of successfully building remarkably complex systems that could not have been built using conventional techniques (DEC, for example, tried three times to build a VAX configuration program using conventional programming approaches prior to initiating the R1 project with John McDermott). They are also building thousands of economically important smaller systems that would have been at least difficult to conceptualize in conventional terms. The second group of activities maintains elements of the pragmatic, domain-specific orientation while explicitly seeking to advance the state of the art by undertaking projects that require a deeper level of complexity and understanding than we currently have. Finally, the third group is working on very specific theoretical issues concerning the nature of computational reasoning and representation. The important point is that the abstract issues are all directly relevant to improving the capabilities of present applications, and, at the same time, the complex applications contribute to understanding the abstract issues by empirically investigating how these abstractions need to be integrated in realistic examples of complex thinking. The field is not a cacophony of disparate activities bearing little relevance to each other. It is rather an attempt to examine simultaneously at several different levels of granularity, the extremely complex and heterogeneous activities of thought. The interplay between these levels of analysis promises to continue to lead to important new computational techniques, from logics and languages, through representational and reasoning techniques, to macro architectures for general reasoning and for reasoning in very complex domains. It promises also to lead to far more precise and fine-grained understanding of intelligence in general and human intelligence in particular. And it promises to enable us to deal far more effectively with the procedures and structures of intelligence as preservable, transferable, and executable entities.

The Skeptical Viewpoint

There are those, of course, who dispute this view and feel that the entire enterprise is ill-advised and doomed to failure. They are often concerned that computation seems capable only of capturing the propositional (i.e., factual) content of human cognition, but not its internal, experiential flavor. By this, they often mean the experience of *qualia*, such as the experience of *greenness* per se, rather than facts about the color green, or other aspects of conscious experience such as pain, emotions, etc. The next part of their argument usually is to the effect that since intelligence clearly involves the interaction of an organism/agent with the world, without this qualitative experience, e.g., consciousness, no agent can really be embedded in the world, and hence without consciousness nothing can be intelligent (i.e., even *act intelligently*). Of the several answers that are possible to these sorts of objections, two seem particularly relevant. One is similar to the usual point regarding digital representations: that they can approximate reality to any degree desired, and, moreover, that in a digital representation one can be certain regarding the size of error that is permitted—a point much more problematic in the case of representing multiple qualitative dimensions rather than just a single numerical value, but one that still seems to carry force. If we are concerned not with the experience of *qualia* but with the implications they have for the actions and inferences a system would take, there does not seem to be any principled reason that representations could not be developed for these entities, and procedures developed that lead a system to act appropriately (e.g., a concept such as "fear" could have procedures that determined which sorts of circumstances would produce it in varying degrees, and also how it would influence the activities of the system. In fact, Simon [1967] argued that any intelligent system approaching the complexity of the human cognitive system would require such a complex set of heterogeneous goals that the organized manipulation and coordination of these goals would require something at least analogous to human emotions).

Artificial Intelligence: The Medium is the Message

Artificial Intelligence as a Representational Medium

The second answer to the skeptics nicely sidesteps the issue of the relationship between consciousness and intelligence, thereby promising to reduce a great deal of largely futile debate (Hill, 1989). Hill suggests that, in artificial intelligence, we are essentially developing a new representational medium, which, like all media, has two facets. The inner or technical facet is that which the creator is aware of during the creation, as when a painter worries about the way that areas of color are applied to a canvas. In this mode, we are all aware, for instance, of the fact that motion pictures consist of a series of still photographs that are shown in rapid succession on a screen, and, on another level, that these still pictures are often created by means (e.g., cartoon drawings, use of actors, use of special effects, etc.) which intentionally ensure that the experience that they produce in the audience will be quite different from the actual reality that was involved in their creation. The outer facet concerns the experience that the audience (including the creator) shares through the medium, as when we suspend our knowledge that the characters of a novel are fictitious, or that the world in front of us is only light on a screen, and relive, as if it were real, the world that is *re-presented* to us.

Artificial intelligence shares this dual nature. When one focuses on the tools of the trade, the techniques and processes by which programs are formed, it is easy to doubt that this is the fundamental stuff out of which real intelligence, let alone consciousness, could be formed. But when one steps back and watches a program performing what appears to be an intelligent action, it is just as easy to suspend our technical knowledge and participate in the experience of perceiving an intelligent entity. (The fact that people are so easy to fool into thinking that a program is behaving intelligently when, in fact, it may be making use of relatively trivial programming devices is an important one from the perspectives of human/machine interaction and program evaluation, but it has absolutely nothing to do with the point at hand—although some critics of artificial intelligence seem to have made careers out of this observation!) In short, with artificial intelligence we are learning how to *re-present* to ourselves the very processes of intelligence, so that we can store them, transmit them, and share them with others, much as we do now with other representational media. As is the case with other media, some of the products will be more useful than others. Some will reflect more accurately what it means to be human than will others. But if the products developed in this media are no more *real* than the products developed in words, music, and visual images, that should be real enough for even the most skeptical critics.

REFERENCES

Allen, J. F. (1984). Towards a general theory of action and time. *Artificial Intelligence, 23*(2), 123-154.

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Ashley, K. D., & Rissland, E. L. (1988). Waiting on weighting: A symbolic least commitment approach. In *AAAI-88: Proceedings of the Seventh National Conference on Artificial Intelligence* (21-26 August 1988) (Vol. 1, pp. 239-244). San Mateo, CA: Morgan Kaufmann.

Battin, P. (1989, April). Talk presented at the School of Library and Information Science, University of Pittsburgh.

Cerf, V. (1989). Talk presented at Carnegie-Mellon University.

Charniak, E. (1982). Context recognition in language comprehension. In W. G. Lehnert & M. H. Ringle (Eds.), *Strategies for natural language processing* (pp. 435-454). Hillsdale, NJ: Lawrence Erlbaum Associates.

Charniak, E. (1988). Motivation analysis, abductive unification, and nonmonotonic equality. *Artificial Intelligence, 34*(3), 275-295.

Cohn, A. G. (1989). Approaches to qualitative reasoning. *Artificial Intelligence Review, 3*(2-3), 177-232.

Davis, R. (1984). Diagnostic reasoning based on structure and behavior. *Artificial Intelligence, 24*(3), 347-410.

Davis, R., & Lenat, D. B. (1982). *Knowledge-based systems in artificial intelligence. Part I—AM: Discovery in mathematics as heuristic search*. New York: McGraw-Hill.

de Kleer, J., & Brown, J. (1984). A qualitative physics based on confluences. *Artificial Intelligence, 24*(1-3), 7-83.

DeJong, G. F., & Mooney, R. (1986). Explanation-based learning: An alternative view. *Machine Learning, 1*(2), 145-176.

Doyle, J. (1979). A truth maintenance system. *Artificial Intelligence, 12*(3), 231-272.

Durfee, E. H. (1988). *Coordination of distributed problem solvers*. Boston, MA: Kluwer Academic Publishers.

Etherington, D. W. (1987a). Formalizing nonmonotonic reasoning systems. *Artificial Intelligence, 31*(1), 41-85.

Etherington, D. W. (1987b). More on inheritance hierarchies with exceptions: Default theories and inferential distance. In *AAAI-87: Proceedings of the Sixth National Conference on Artificial Intelligence* (13-17 July 1987) (Vol. 1, pp. 352-357). Los Altos, CA: Morgan Kaufmann.

Falkenhainer, B.; Forbus, K. D.; & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence, 41*(1), 1-63.

Fikes, R. E., & Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence, 2*(3/4), 189-208.

Forbus, K. D. (1984). Qualitative process theory. *Artificial Intelligence, 24*(1-3), 85-168.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science, 7*(2), 155-170.

Gentner, D., & Stevens, A. L. (Eds.). (1983). *Mental models*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Georgeff, M. P., & Ingrand, F. F. (1989). Decision-making in an embedded reasoning system. In *IJCAI-89: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (20-25 August 1989) (Vol. 2, pp. 972-978). San Mateo, CA: Morgan Kaufmann.

Ginsberg, M. L. (Ed.). (1987). *Readings in nonmonotonic reasoning*. Los Altos, CA: Morgan Kaufmann.

Greiner, R. (1988). Learning by understanding analogies. *Artificial Intelligence, 35*(1), 81-125.

Hasling, D. W.; Clancey, W. J.; & Rennels, G. (1984). Strategic explanations for a diagnostic consultation system. *International Journal of Man-Machine Studies, 20*(1), 3-19.

Hayes-Roth, B., & Hayes-Roth, F. (1979). A cognitive model of planning. *Cognitive Science, 3*(4), 275-310.

Hill, W. (1989, June). The mind at AI: Horseless carriage to clock. Paper presented at the Fourth Annual Rocky Mountain Conference on Artificial Intelligence.

Horty, J. F., & Thomason, R. H. (1988). Mixing strict and defeasible inheritance. In *AAAI-88: Proceedings of the Seventh National Conference on Artificial Intelligence* (21-26 August 1988) (Vol. 2, pp. 427-432). San Mateo, CA: Morgan Kaufmann.

Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness*. Cambridge, MA: Harvard University Press.

Kass, A. (1989). Adaptation-based explanation: Extending script/frame theory to handle novel input. In *IJCAI-89: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (20-25 August 1989) (Vol. 1, pp. 141-151). San Mateo, CA: Morgan Kaufmann.

Kolodner, J. L. (1984). *Retrieval and organizational strategies in conceptual memory: A computer model*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kolodner, J. L. (Ed.). (1988). *Case-based reasoning* (Proceedings of a DARPA Case-Based Reasoning Workshop, 10-13 May 1988). San Mateo, CA: Morgan Kaufmann.

Krawchuk, B. J., & Witten, I. H. (1989). Explanation-based learning: Its role in problem solving. *Journal of Experimental & Theoretical Artificial Intelligence, 1*(1), 27-49.

Kuipers, B. (1984). Commonsense reasoning about causality: Deriving behavior from structure. *Artificial Intelligence, 24*(1-3), 169-203.

Kuipers, B. (1986). Qualitative simulation. *Artificial Intelligence, 29*(3), 289-338.

Kulkarni, D., & Simon, H. A. (1988). The processes of scientific discovery: The strategy of experimentation. *Cognitive Science, 12*(2), 139-175.

Laird, J. E.; Newell, A.; & Rosenbloom, P. S. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence, 33*(1), 1-64.

Lenat, D. B. (1983). Theory formation by heuristic search. *Artificial Intelligence, 21*(1, 2), 31-59.

Lenat, D. B., and Guha, R. V. (1990). *Building large knowledge-based systems: Representation and inferences in the CYC project.* Reading, MA: Addison-Wesley.

Lesser, V. R.; Corkill, D. D.; Whitehair, R. C.; & Hernandez, J. A. (1989). Focus of control through goal relationships. In *IJCAI-89: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (20-25 August 1989) (Vol. 1, pp. 497-511). San Mateo, CA: Morgan Kaufmann.

Lewis, C. (1988). Why and how to learn why: Analysis-based generalization of procedures. *Cognitive Science, 12*(2), 211-256.

Lucky, R. (1989). Information as seen through the eyes of humans and machines. Samuel Lazerow Memorial Lecture, University of Pittsburgh, November 17, 1989.

McCarthy, J. (1980). Circumscription—A form of non-monotonic reasoning. *Artificial Intelligence, 13*(1, 2), 27-39.

McCarthy, J. (1986). Applications of circumscription to formalizing common-sense knowledge. *Artificial Intelligence, 28*(1), 89-116.

McCarthy, J., & Hayes, P. J. (1969). Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer, D. Michie, & M. Swann, *Machine Intelligence 4* (pp. 463-502). New York: American Elsevier.

McDermott, D. (1982). A temporal logic for reasoning about processes and plans. *Cognitive Science, 6*(2), 101-155.

Metzler, D. P. (1989). Design requirements and an architecture for intelligent interactive workbenches. In *ASIS '89: Managing information and technology* (Proceedings of the 52nd ASIS Annual Meeting, 30 October-2 November 1989) (pp. 66-70). Medford, NJ: Learned Information.

Metzler, D. P. (1990). Connectionist and symbolic information processing: A critical analysis and suggested research agenda for connectionism from the symbolic perspective. In D. Henderson (Ed.), *ASIS '90: Information in the year 2000: From research to applications* (Proceedings of the 53rd ASIS Annual Meeting, 4-8 November 1990) (pp. 261-278). Medford, NJ: Learned Information.

Metzler, D. P., & Williams, J. (1988). ISLAND: An intelligent system for local area network design. In Z. W. Ras & L. Saitta (Eds.), *Methodologies for intelligent systems, 3* (Proceedings of the Third International Symposium on Methodologies for Intelligent Systems, 12-15 October 1988) (pp. 122-131). New York: North Holland.

Mitchell, T. M. (1977). Version spaces: A candidate elimination approach to rule learning. In *IJCAI-77: Proceedings of the Fifth International Joint Conference on Artificial Intelligence* (22-25 August 1977) (Vol. 1, pp. 305-310). Los Altos, CA: William Kaufmann.

Moore, J. D., & Swartout, W. R. (1988, August). A reactive approach to explanation. Paper presented at the AAAI Workshop on Explanation.

Newell, A., & Simon, H. A. (1976). Computer science as empirical inquiry: Symbols and search (Tenth Turing Award Lecture). *Communications of the ACM, 19*(3), 113-126.

Norman, D. A. (1983). Some observations on mental models. In D. Gentner & A. L. Stevens (Eds.), *Mental models* (pp. 7-14). Hillsdale, NJ: Lawrence Erlbaum Associates.

Norvig, P. (1989). Marker passing as a weak method for text inferencing. *Cognitive Science, 13*(4), 569-620.

Reiter, R. (1978). On closed world data bases. In H. Gallaire & J. Minker (Eds.), *Logic and data bases* (pp. 55-76). New York: Plenum.

Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence, 13*(1, 2), 81-132.

Reiter, R. (1987). A theory of diagnosis from first principles. *Artificial Intelligence, 32*(1), 57-96.

Rissland, E. L., & King, J. A. (Eds.). (1988, August). Paper presented at the Case-Based Reasoning Workshop. American Association for Artificial Intelligence.

Rissland, E. L., & Skalak, D. B. (1989). Combining case-based and rule-based reasoning: A heuristic approach. In *IJCAI-89: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (20-25 August 1989) (Vol. 1, pp. 524-531). San Mateo, CA: Morgan Kaufmann.

Schank, R. C. (1982). Reminding and memory organization: An introduction to MOPs. In W. G. Lehnert & M. H. Ringle (Eds.), *Strategies for natural language processing* (pp. 455-493). Hillsdale, NJ: Lawrence Erlbaum Associates.

Schneider, W. (1987). Connectionism: Is it a paradigm shift for psychology? *Behavior Research Methods, Instruments & Computers, 19*(2), 73-83.

Shoham, Y. (1987). Temporal logics in AI: Semantical and ontological considerations. *Artificial Intelligence, 33*(1), 89-104.

Shoham, Y., & McDermott, D. (1988). Problems in formal temporal reasoning. *Artificial Intelligence, 36*(1), 49-61.

Shortliffe, E. H. (1976). *Computer-based medical consultations: MYCIN*. New York: Elsevier.

Simon, H. A. (1967). Motivational and emotional controls of cognition. *Psychological Review, 74*(1), 29-39.

Smith, B. C. (1982). *Reflection and semantics in a procedural language* (Unpublished doctoral dissertation and Technical Report MIT/LCS/TR-272). Cambridge, MA: Massachusetts Institute of Technology.

Sycara, K. (1988). Resolving goal conflicts via negotiation. In *AAAI-88: Proceedings of the Seventh National Conference on Artificial Intelligence* (21-26 August 1988) (Vol. 1, pp. 245-250). San Mateo, CA: Morgan Kaufmann.

Talmy, L. (1988). Force dynamics in language and cognition. *Cognitive Science, 12*(1), 49-100.

Tanca, L., & Ceri, S. (1986). ELAND: An expert system for the configuration of local area networks' applications. In *Proceedings of the 11th Conference on Local Computer Networks* (6-8 October 1986) (pp. 89-98). Washington, DC: IEEE Computer Society.

Touretsky, D. S. (1986). *The mathematics of inheritance systems*. London: Pitman.

Tulving, E. (1972). Episodic and semantic memory. In E. Tulving & W. Donaldson (Eds.), *Organization of memory* (pp. 381-403). New York: Academic Press.

Wallis, J. W., & Shortliffe, E. H. (1982). Explanatory power for medical expert systems: Studies in the representation of causal relationships for clinical consultations. *Methods of Information in Medicine, 21*(3), 127-136.

Wick, M. R., & Thompson, W. B. (1989). Reconstructive explanation: Explanation as complex problem solving. In *IJCAI-89: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (20-25 August 1989) (Vol. 1, pp. 135-141). San Mateo, CA: Morgan Kaufmann.

Wilensky, R. (1983). *Planning and understanding: A computational approach to human reasoning*. Reading, MA: Addison-Wesley.

Winston, P. H. (1975). Learning structural descriptions from examples. In P. H. Winston (Ed.), *The psychology of computer vision* (pp. 157-209). New York: McGraw-Hill.

CHARLES FENLY*

Assistant Chief, Descriptive Cataloging Division
Library of Congress
Washington, DC

Technical Services Processes as Models for Assessing Expert System Suitability and Benefits

ABSTRACT

The rudiments of a strategy for assessing the appropriateness of using an expert system in a given domain are presented. The assessment process involves comparing the characteristics of the domain with various suitability criteria and identifying potential benefits from an expert system in the domain. Two library technical service functions, descriptive cataloging and shelflisting, are used as models illustrating this assessment process. Based on organizational factors specific to the Library of Congress, series work (a subset of descriptive cataloging) and shelflisting appear to be suitable and beneficial candidates for expert system development efforts.

INTRODUCTION

Two questions that might be posed about the introduction of expert systems technology into the technical service workplace are: *Can* expert systems be applied to library technical service processes? and, if so, *Should* expert systems be applied to library technical service processes? Of these, the second is by far the more interesting and challenging question.

---

*The views expressed are those of the author and do not reflect the official policies of the Library of Congress.

The first question may be answered fairly easily. An examination of the literature of expert systems suggests rather clearly that, given the will, time, and resources, expert systems could be applied in some way to such activities as cataloging, classification, acquisitions work, serials management, and the like.

But even if it is accepted that expert systems could be implemented in technical services operations, the question remains, should they be? There is no single, conclusive answer to this question. However, there are strategies for approaching this question that an organization can use to help make rational decisions about whether an expert system has a place in its own operations. This paper will present some of the fundamentals of such a strategy.

DECIDING ON AN EXPERT SYSTEM

As a technical services manager, this author believes strongly that decisions related to implementation of expert systems technology in the technical services (or any other) workplace should be based on sound management decision making. This may sound obvious, and perhaps it would be obvious if the topic of discussion were something other than an aspect of artificial intelligence (AI) technology. But AI technology carries with it such a degree of fascination and, one might even say, glamour, that the possibility of trying to introduce such technology for its own sake rather than for sound management reasons is present with respect to expert systems in ways that might not be the case with other technologies.

What is it that prompts a consideration of using an expert system in technical processing? First, there is an increasingly wide awareness among technical services librarians that computer programs have been developed which exhibit human-like reasoning, which may be able to learn from their mistakes, and which quickly and cleverly perform tasks normally done by scarce and expensive human experts. Further, it is widely recognized that automation has paid off in a big way in technical processing operations in the past: through creative use of computing, marvels of information storage and retrieval and resource sharing have been achieved. It is therefore natural that technical services librarians would wish to assess whether this newer technology has the potential to confer similar benefits.

Upon further investigation of the matter in the literature, an increased understanding of the realities and limitations of expert systems technology and a greater awareness of what expert systems can and cannot do might lead to some decline in enthusiasm. While it may be true that expert systems have been programmed to solve complex problems, this has tended to require considerable expenditure of time and money, and even after such expenditure, many systems have never gone into production because of reliability problems. And though research is underway to improve the processes by which systems learn, at present, the acquisition of knowledge by an expert system is one of the most difficult aspects of system development—a major bottleneck rather than one of the strengths of the technology (Rolston, 1988, pp. 157-67). Nevertheless, there is something rather compelling about this technology, so that even while recognizing that it is not a panacea, we may continue to have a strong interest in examining more closely whether there are prospects for using expert systems in our organizations.

According to Greene (1990, pp. 48-59), artificial intelligence technology has become so well integrated in some Japanese organizations that it can function as a frequently used tool of problem solving and work improvement. Clearly, this is not the situation that prevails in libraries today. Artificial intelligence technology is relatively unknown and may even be viewed as somewhat exotic, and an effective way to get to know it better is to investigate its potential usefulness.

A much less defensible approach to learning about expert systems, however, is to embark hastily upon an expert systems development project based on the premise: "Let's think of something that we can develop an expert system to do." The objection to this approach is a practical one: It is too likely to result in projects that go nowhere, in systems that do not produce useful results, that is, "toy systems."

This does not imply that there is anything wrong with developing a small expert system. There may be definite benefits to be gained from implementing a small system which deals effectively with a real problem which happens to be small in size. As our familiarity with the technology increases, and as more powerful and user-friendly development tools become available, it may become increasingly common for domain experts to engage in their own knowledge engineering to develop such systems to help them do their work. But this appropriate use of expert systems technology is quite different from projects whose end result is a "demonstration prototype": a small system which is small because (a) it deals with a tiny piece of a large domain, with no clear plans to expand it to the point where it can address a meaningful subset of the domain; or (b) it is a shallow and superficial cut at a deep and complex problem. Projects such as these do not confer upon an organization the kind of benefits which expert systems have the potential to yield.

There are no doubt many different strategies which could be proposed for assessing the appropriateness of implementing an expert system in a given domain in a particular organization. The discussion which follows will suggest one possible approach. Though the approach described has general applicability, the discussion will relate the issues addressed to specific technical processing activities.

Assessing the Appropriateness of an Expert System

In 1987, Howard Harris and I conducted an investigation of the feasibility of applying expert systems to technical processing operations in Processing Services (now known as Collections Services) of the Library of Congress (Fenly & Harris, 1988). Since neither of us was an expert system expert, we began with an extensive literature review to gain a better understanding of the technology. That survey of the literature convinced us that expert systems potentially have great power. But it also convinced us that genuine expert systems, with the depth and power to solve substantial and meaningful problems, are time consuming and costly to develop and that expert system development projects have uncertainties associated with them that would probably not be tolerated in connection with traditional data processing initiatives. Thus, it was clear that there have to be other reasons for implementing an expert system besides the fact that it would be intellectually stimulating to do so.

And, in fact, there are other reasons: namely, the potential benefits to be derived from a successful implementation of an appropriate expert system. Some of the potential benefits of expert systems that have been described in the literature are these:

- expert systems can make scarce expertise more widely available within the organization, thereby helping nonexperts achieve expert-like results;
- they can free human experts for other activities besides repeatedly solving the problems which an expert system could address;
- they can promote standardization and consistency in the solving of relatively unstructured tasks;
- they can enhance organizational effectiveness and efficiency by making readily available solutions to difficult problems which might otherwise require time-consuming research or consultation with experts to solve;
- they can provide a means for capturing and storing valuable knowledge that might be lost if an employee with scarce expertise left the organization;
- they can provide a means for long-term retention of complex knowledge, since machine knowledge does not deteriorate with time or disuse in the same way that human knowledge does;
- they can perform, at a consistently high level, tasks which humans might perform inconsistently due to fatigue or loss of concentration (Beerel, 1987, pp. 84-85; Olsen, 1989, pp. 121-22; Waterman, 1986, pp. 12-13).

As stated above, these are potential benefits to be derived from successful implementation of an *appropriate* expert system. That raises another question: What are the criteria for assessing whether a particular domain is *suitable* for an expert system?

Assessing the Suitability of an Expert System

The following list of suitability criteria (a slightly modified version of the list used in the 1987 Library of Congress study) was based on work by Prerau (1985). It must be emphasized that the list given here is only representative, not exhaustive. It is intended to give a flavor of the characteristics of an expert system type of problem. For a detailed and comprehensive discussion of this important topic, see Prerau (1990).

Note that these are referred to as "suitability" criteria, not "feasibility" criteria. The fact that it might be feasible to apply AI programming techniques to a problem does not in itself make that problem a suitable domain for an expert system. Conventional data processing techniques or nonautomated tools such as manuals, flowcharts, or decision logic tables may be more appropriate ways of addressing a given problem or task.

Selected Domain Suitability Criteria

1. Tasks to be performed and problems to be solved in the domain require expert knowledge, judgment, and experience. In other words, problems in the domain are nontrivial, and experienced people perform the work at a significantly higher level than novices.
2. Tasks and problems in the domain require primarily symbolic (rather than algorithmic) reasoning and require the use of heuristics. Otherwise, more familiar and possibly more efficient conventional data processing techniques might be more appropriate.
3. Tasks and problems in the domain have appropriate depth. In practice, tasks to be performed might typically take an expert a few minutes to a few hours to perform. A domain which lacks depth is not a good expert system domain for at least two reasons: First, if tasks typically take only seconds to perform, the work might actually be slowed down by the time required to interact with an expert system; second, users are likely to become bored with and discontinue using a system that answers only simple questions. On the other hand, if a domain is so deep that tasks take many hours to complete, an expert system in the domain might be unmanageably large and unacceptably slow and expensive.
4. The domain is relatively narrow, well bounded, and self-contained. Since an expert system should deal with problems of meaningful depth, a domain which is extremely broad or unbounded could

overwhelm a development effort. This potential problem can be mitigated, however, if a large domain can be segmented into manageable parts.

5. Some degree of incorrect or nonoptimal results can be tolerated. This is important because expert systems are subject to producing unreliable or invalid results due to certain inherent limitations of the technology (Waterman, 1986, p. 29; Hollnagel, 1989, pp. 33-35). Furthermore, expert system knowledge engineering is subject to the law of diminishing returns to scale. That is, the incremental utility of the system increases by ever smaller amounts, as additional costs are incurred to improve system performance by increasing the percentage of domain knowledge embodied in the system. Eventually, a point will be reached where the marginal cost of adding further knowledge will exceed the resulting marginal increase in the usefulness of the system, at which point knowledge acquisition should cease. The system will contain less than complete domain knowledge at this point (Kang & Levy, 1989, pp. 242-43).
6. The domain is fairly stable, with the need for the task projected to continue for several years, with changes tending to be gradual and evolutionary, and with no radical changes which would redefine the task being planned. This is important because of the anticipated amount of time associated with development of a substantial system. It might be very hard to evaluate during the development process the performance of a system in a highly volatile domain. Needless to say, it is important that the system still be useful and relevant at the time it is ready to go into production.
7. There are recognized experts working in the domain who would be willing and available to participate in a development project. These experts would normally be the principal source of the expertise which is to be embodied in the system.

As already stated, the above list is not exhaustive, but it does provide some essential considerations in determining whether a task is an expert systems kind of task or whether it might better be dealt with through other approaches, such as manual processes or conventional algorithmic automated data processing.

Organizational Factors to be Considered

The criteria just discussed focus on the technical suitability of a domain as a potential expert systems candidate. In any given organization, there will also be organizational factors which would have to be considered before deciding whether to embark upon a development project. These might include such considerations as the following:

1. Is management supportive of a development project? Will management fund the project or support the seeking of funds from other sources? Can the organization's hardware, software, and professional time be devoted to a development effort?
2. Are there political objections to development of a system in the domain under consideration? Will management and staff working in that domain feel threatened or intimidated by the expert system and resist it?
3. Is there organizational support for maintaining a system once implemented? Even a seemingly stable domain might turn out, upon closer inspection, to be more volatile than one might have thought. An unmaintained system might soon begin giving wrong answers and could then be expected to fall into disuse.

It is obvious that the "wrong" answer to one or more of these questions could make a development project untenable.

DESCRIPTIVE CATALOGING AND SHELFLISTING

Now that some of the potential benefits which an organization might derive from expert systems and some criteria for determining the characteristics of an appropriate candidate for an expert system have been presented, two traditional technical services functions will be used as models of how a process might be examined against the suitability criteria and benefits. These two functions are descriptive cataloging and shelflisting. They were chosen as the models for this discussion largely because they are presumed to be more generally familiar than many other technical services functions, such as acquisitions and serials management, which may be performed rather differently at different organizations. It should nevertheless be emphasized that decisions about expert system development have to take the environment of the specific organization where they are intended to be used into account, and that will be reflected in what follows. In the present discussion, the organizational realities are those of the Library of Congress.

Descriptive Cataloging as an Expert System Domain

Descriptive cataloging is the subset of cataloging activity which involves (1) providing a bibliographic description of an item sufficient to identify the item and to provide to a prospective user certain information necessary to make judgments about its usefulness, and (2) formulating uniform access points to enable the potential user to retrieve the bibliographic record.

An examination of the library and information science literature reveals a definite interest in considering the application of expert systems technology to the rules and procedures of descriptive cataloging. This is hardly surprising, since the most common expert system knowledge base building block is the rule, and descriptive cataloging is certainly rule oriented; indeed, with its basis in the *Anglo-American Cataloguing Rules,* second edition *(AACR2)* (Gorman & Winkler, 1988), it is one of the most codified domains in librarianship. A particular focus of interest for purposes of suggesting hypothetical expert systems or building prototype systems has been chapter 21 of *AACR2,* which deals with the process of choosing access points. Upon cursory examination, this chapter appears to lend itself to the formulation of many rules in the form exemplified by the following:

> IF court rules govern a single court
> THEN main entry is the heading for the court (Rule 21.34A, Modified from *AACR2,* 1988, p. 364)

Attempts to develop a knowledge base built in such a fashion have tended, however, to produce unconvincing results. An example of an *AACR2*-based system which has been described in the literature is CATALYST (Gibb & Sharif, 1988). This system was developed using the PC-based expert system shell, ESP-Advisor. A feature of this shell, called "text animation," facilitates the conversion of existing documentation into an expert system knowledge base. CATALYST works by presenting the user with various menus; the user is expected to indicate a choice, which the system then uses to consult the knowledge base and either advance to the next level in the decision tree or provide an answer to the problem being addressed. In the report on this system, several examples of such menus relating to choice of main entry heading are provided. An examination of these examples prompts questions about the probable usefulness of this system. It seems likely that the appropriate menu choice will often not be evident to the novice with limited cataloging knowledge. Though there is a help facility, the decision to consult it depends on recognizing what one does not know; this is often far from obvious when an inexperienced person is dealing with such complex matters as choosing a bibliographic access point. And the experienced cataloger, if he or she does not already know the right answer, will probably want to read the rules carefully in order to understand the correct approach in its context, as opposed to relying on the skeletal information provided by the help facility.

The problem at work here is one that several writers have pointed out: The expertise in cataloging is not explicit in the rules; rather, it is implicit in the heuristics employed by the experts who do the work (Davies, 1986, p. 72). Consulting *AACR2* is not synonymous with descriptive cataloging: "Like most professional handbooks, it is written for those who already know" (Hjerppe et al., 1985, p. 12). In fact, this problem is noted in the report on CATALYST, making slightly puzzling the authors' conclusion that "[CATALYST's] value as an assistant is yet to be assessed but it seems likely that it can contribute to both educational and operational environments" (Gibb & Sharif, 1988, p. 70). Another possible conclusion might have been that development of an expert system in descriptive cataloging which possesses genuine expertise would require very extensive knowledge engineering, and is therefore a problem of a completely different order of magnitude from that of using an expert system shell to recast the cataloging rules into an automated format.

Thus, the appropriateness of applying an expert system to a particular domain should not be assumed too hastily. Descriptive cataloging is rule based and expert systems are frequently rule based, but this apparent similarity is by no means adequate evidence that the descriptive cataloging rules constitute a suitable expert systems domain. A decision that a domain is right for an expert system is better arrived at through a careful comparison of the characteristics of that domain to suitability criteria such as those discussed above.

In the Library of Congress study (Fenly & Harris, 1988), such comparisons were made in a number of domains. The following is an example of the results of such a comparison with respect to the domain of descriptive cataloging.

1. Do the tasks to be performed and problems to be solved in this domain require expert knowledge, judgment, and experience? This question can be answered confidently in the affirmative. There are marked differences in performance between the novice and the experienced individual in this domain, and the time required to achieve performance levels characteristic of the best practitioners is likely to be measured in years. Thus, this is an expert domain.
2. Do the experts in this domain use symbolic reasoning and heuristic problem solving? Again, the answer is yes. This is particularly the case in subsets of the domain involving complex relationships or research, such as series work or work involving formulation of complex name headings or uniform titles.
3. Do the tasks to be performed possess the desired degree of depth? The answer here is not so obvious. Although the full process of completing the descriptive cataloging portion of a particular bibliographic record might fit neatly into the "few minutes to few hours" time frame, the process in practice consists of a number of discrete steps, and many of the necessary decisions are usually made by an experienced individual almost as quickly as he or she can examine the item being cataloged, and certainly in less time than would be required to interact with an expert system. Certain

subtasks of descriptive cataloging, however, are intricate enough in themselves to satisfy this criterion. A good example is series work. Since this subset of the descriptive cataloging domain will be the focus of further attention later in this paper, a brief discussion of series work and what a series expert system might do will be presented at this point.

A Series Expert System

The Fenly and Harris investigation at the Library of Congress suggested that series work is the aspect of descriptive cataloging most likely to require a disproportionate amount of consultation to resolve unusual problems. In fact, such consultation was involving so much of the attention of certain experts in the Office for Descriptive Cataloging Policy that the office embarked on a special training program to increase the number of series experts within the monographic cataloging sections. Several factors make series work uniquely challenging, including the problems of seriality, the number and complexity of series-related rules and procedures, and the difficulties that stem from the need to relate newly received items to existing series, many of which were established under different rules and practices from those now in place.

An expert system which would help address these problems would include the knowledge and heuristics which the best experts apply to deal with these troublesome matters. The system would assist the user in pinpointing the nature of the problem, perhaps through the use of increasingly detailed levels of menus. It would be capable of asking for information needed to evaluate the problem, and it would be able to recommend a solution or recognize that it lacked adequate knowledge to solve the problem. As a by-product of containing the facts and heuristics associated with series work, it would be capable of assisting in the establishment of a new series, including determination of proper form of headings, references, and treatment.

4. Is the task relatively narrow, well bounded, and self-contained? Our investigation at the Library of Congress convinced us that the domain of descriptive cataloging as a whole is much too broad for an expert system which attempts to cover the full range of tasks at an adequate level of depth to be appropriate. It is therefore important to subdivide this domain in order to focus on a narrow subset of problems so that a realistically deep expert system can be contemplated. Series work constitutes such a subdivision.
5. Can some degree of incorrect or nonoptimal results be tolerated? Traditionally, a high degree of accuracy in adherence to cataloging rules and procedures has been considered the norm. In the present environment of automated storage and retrieval of bibliographic information, accuracy and consistency are as important as ever. An

expert system that delivered wrong answers too often would thus be unacceptable. Unfortunately, as noted above, expert systems are subject to the law of diminishing returns with respect to fine-tuning their level of performance beyond a certain point. This poses a challenge to the would-be developer of an expert system in cataloging. Unless the system can be fine tuned to yield results of acceptable accuracy, it will either never be implemented or will quickly fall into disuse. It therefore becomes important to ask the question: What is an acceptable performance level? Rolston (1988, pp. 213-15) provides a useful perspective on this question. A primary purpose of an expert system is to distribute an expert's knowledge to non-expert users. Therefore, a system's effectiveness should be evaluated not by comparing its results to some theoretical model of perfection but by comparing its performance to what the intended users would achieve without the system's help. Viewed from this perspective, it is reasonable to assume that an expert system of an acceptable performance level could be developed in the domain of series work, though it may be hard to judge in advance how much effort would be required to attain that level.

6. Is the domain fairly stable? Are significant changes anticipated in the near future? These are most important questions because of the anticipated length of time required to bring into production a substantial expert system application. Some years ago, it would have seemed rather obvious that this was a reasonably stable domain. At present, however, it appears that environmental forces, chiefly economic, may have the effect of introducing increased volatility into this domain. In the face of budgetary constraints leading to reduced staff levels and growth of backlogs of uncataloged materials, serious attention is being given to descriptive cataloging simplification. This could have the effect of bringing about changes to existing practices, which could significantly complicate a system-development effort mounted in the near future.
7. Are there recognized experts working in the domain today? There are indeed recognized and articulate experts available to lend their knowledge and experience to a development effort.

The process of evaluating the domain of descriptive cataloging against the suitability criteria thus yields somewhat mixed results. A number of the criteria appear to be well satisfied, with those relating to appropriate task depth and domain breadth seeming to be satisfied best by one of the more complex subsets of the domain, such as series work. On the other hand, due to the diminishing returns problem in connection with expert systems development, it may be hard to predict in advance how much effort (and therefore cost) will be required to implement a system which will demonstrate acceptable accuracy levels, and any development effort mounted in the near future might be subject to being hampered by possible changes in practice in the domain.

On balance, if it is assumed that these two concerns can be satisfactorily addressed, it could reasonably be concluded that a complex subset of descriptive cataloging such as series work does appear to be a suitable expert system domain. In the course of the 1987 investigation, we concluded that series work was in fact one of the domains which, from among all the technical processing operations we investigated, seemed best to satisfy the suitability criteria.

If a domain seems suitable, it must then be determined whether implementing a system in that domain is likely to yield any benefits. There does appear to be the potential for benefits from a series expert system, including the following:

- As noted above, series expertise is scarce, and the system could be expected to make this scarce expertise more widely available.
- The system would free human series experts from repeatedly solving difficult series problems, thereby allowing them to turn their attention to other matters for which they are responsible.
- The amount of time-consuming research and consultation in an effort to resolve series problems should be reduced.
- Valuable knowledge and heuristics related to resolving series problems would be retained in an expert system and would continue to benefit the organization even if a human expert resigned or retired.

Since these are obviously significant benefits and since series work appears to be a suitable domain for an expert system, it would appear that this is an application worthy of serious consideration for a development effort. There is, however, one more crucial matter to consider: cost. That topic will be addressed below. First, the other major technical services function to be examined in detail in order to consider its suitability and benefits as an expert systems domain will be discussed. That function is shelflisting.

Shelflisting as an Expert System Domain

Because of the large volume of work passing through the cataloging and classification workstream, shelflisting at the Library of Congress is done by a separate section of more than sixty staff members. The principal intellectual effort of this work entails formulating a book number, known as the cutter number, which is added to the classification number provided by a subject cataloger to produce a call number unique to the item in hand. Though the cutter number is based on a simple table, in practice, the work is complicated by two factors. First, the classification schedules, which prescribe how the call number is to be structured, are extensive and complex. Not every classification number is completed according to the same formula. Second, because of the immense size of the existing shelflist, a cutter number derived from the cutter table can only be suggestive. The task of finally formulating the book number takes place at the shelflist itself, where the shelflister must fit the item now being processed into what has already been done.

Does this task constitute a suitable expert system domain? In considering that question, a conceptual model of an expert system-based approach to shelflisting is helpful. Such an approach might be based upon an expert system interacting with a database of shelflisting records. These records would contain the call number and the subset of the fields contained in a full MARC record on which the formulation of the call number depended (and, to permit fully automated shelflisting, fields for holdings information). This database might reside on a minicomputer or on CD-ROM supplemented with a dynamic database of shelflisting records formulated since the most recent issue of the CD-ROM file.

The expert system component would contain rules specifying how the cutter number should be derived in the case of each unique method of cuttering. Each rule would be linked to a database of classification numbers whose cutters are to be derived according to that rule. Thus, when the operator, in response to the system prompt, keyed in the classification number, the system would know which rule applied and could then ask for any additional data needed. The expert system could then apply its rules for actually formulating the cutter number. As part of this process, the system would consult the shelflisting record database to determine where the new record should fit, determine the correct cutter number based on that fit, formulate the shelflisting record, and add it to the database.

With this model in mind, a comparison of the domain against the list of suitability criteria can be made.

1. Is this a domain which requires expert knowledge, judgment, and experience? Because of the complicating factors already described, this is in fact a domain in which experienced practitioners perform much better than novices. A substantial program of formal training and a lengthy period of experience are required before a shelflister typically reaches a high level of proficiency in dealing with the full range of complex problems.
2. Does the task require symbolic reasoning and the use of heuristics? A superficial examination of the task would suggest that it is largely algorithmic. However, although the use of heuristic problem solving in this domain is not as great as in a domain such as series work,

the level of complexity of the work is such that it cannot be carried out by purely algorithmic procedures.

3. Does the task possess the appropriate level of depth? Because the shelflister must make decisions based upon the complicated and extensive classification schedules and upon the sometimes intricate realities of the shelflist, the task is not a trivial one which can be dispensed with in a few seconds. Thus, this criterion would seem to be satisfied.
4. Is the task relatively narrow, well bounded, and self-contained? Given the size of the classification schedules upon which the system would be dependent, it may be hard to see the domain as narrow. However, each separate shelflisting decision focuses on one small part of the schedules and of the shelflist itself. Furthermore, because of the way the classification schedules are structured, it should be possible to segment the domain for system development. In addition, though there are thousands of classification numbers, there are only a few ways to complete a call number. Thus, the domain appears to be sufficiently narrow.
5. Can some degree of incorrect or nonoptimal results be tolerated? Clearly, it is essential that call numbers be correct in the sense that the number in the cataloging record must match the number that appears on the shelved item. But, perhaps in some other respects, some nonoptimal results could be tolerated. If the number assigned to an item were slightly off the mark (for example, suppose an item by Jones in a given classification should shelve immediately after Johnson but gets put by the system immediately ahead of Johnson), this would certainly not be desirable, but a small number of such misassignments might not be excessively harmful. Furthermore, it is possible to conceive of ways to help prevent an excessive number of errors of this type. For example, two features that might be built into a system to assist in error-prevention are (a) a display of the system's results to the operator in context (for example, a display showing the newly derived shelflisting decision along with the two records that come immediately before and the two that come immediately after it); and (b) the ability to note and call to the operator's attention certain kinds of anomalies (for example, to note that although the rule it is applying calls for single cuttering, other records in that class seem to be double-cuttered).
6. Is the domain stable? It is, since changes in practice tend to be gradual and there are no significant new developments currently being planned.
7. Are there recognized experts working in the domain? There are experts with many years of experience who are articulate, capable of providing authoritative answers to the most difficult of problems, and whose expertise is widely recognized.

The comparison of the domain of shelflisting to the suitability criteria thus suggests that the domain is a potential candidate for an expert system development effort. With respect to potential benefits, there are several which might be anticipated from implementation of a properly functioning system along the lines of the model under discussion:

- If such a system could produce credible results with acceptable consistency, the exceptional labor intensity of the task as it is now constituted could be greatly mitigated. Staff could be redeployed to some of the many other pressing tasks in the organization which are not so amenable to being assisted by technology.
- As the system evolved and heuristics for dealing with some of the more unusual and complex problems were added, the number of time-consuming consultations with the most experienced experts could be lessened.
- The enormous shelflist as it currently exists has been developed over many years and embodies a good deal of implicit knowledge which may be fully understood only by a few individuals with many years of experience working in this area. If such a system could capture this knowledge, the operation would continue to benefit from the experience and expertise of these individuals even after they retired.
- Though this work does require some degree of expertise and heuristic problem solving, it is also production oriented and repetitious, so that the risk of errors and inconsistencies resulting from human fatigue is always present. An expert system would not be subject to this problem.

Thus, there appears to be the potential for truly significant benefits from a system which would function as proposed at an acceptable performance level.

Both series work and shelflisting seem to be suitable and potentially beneficial domains in which to apply expert systems technology. If it is determined that a proposed application is suitable and beneficial, and if it is assumed that organizational factors such as those noted above are not a barrier, should development work then proceed? That is certainly an option available to an organization intent on implementing an expert system. However, from a sound managerial decision-making point of view, a preferable next step would be a careful assessment of costs in relation to expected benefits.

Cost Considerations

No attempt will be made here to suggest a methodology for a cost-benefit analysis. For an organization lacking expertise in knowledge engineering, such an analysis may be difficult or even impossible to conduct "in-house." The literature of AI and expert systems offers little useful information about development costs. An additional complicating factor in a cost-benefit analysis is the intangible nature of some of the benefits sought from an expert system, such as wider dissemination of expertise and the capability to retain scarce knowledge. It may therefore be necessary to bring in a knowledge engineering consultant to assist in the analysis. This could be costly, since the consultant will presumably have a great deal to learn about the domain in order to offer sound judgments about how challenging the development effort is likely to be in order to achieve the hoped-for benefits.

Despite the difficulties, the alternative to such a cost-benefit analysis would be to proceed into a realm of considerable uncertainty. If it is true that a "small, fairly uncomplicated system" may cost $40,000 to $100,000, and that the cost of a large-scale system developed on a mainframe could exceed $1 million (Beerel, 1987, p. 61), it would seem highly advisable to undertake a development effort with the clearest possible idea in mind of what results are expected and what level of effort is likely to be required to achieve those results.

CONCLUSION

This paper has attempted to present the rudiments of a rational, businesslike strategy for identifying promising candidates for the application of expert systems technology. Two traditional and well-known library technical services functions were used as models to illustrate how such a strategy might work. Based on circumstances specific to the Library of Congress, series work (a subset of the larger domain of descriptive cataloging) and shelflisting appeared to be promising candidates for expert systems based on considerations of domain suitability and potential benefits (and pending a favorable cost-benefit analysis). No conclusions can be drawn from the foregoing about the applicability of expert systems to technical processing generally, however, since the appropriateness of implementing an expert system depends on so many organization-specific factors.

REFERENCES

Beerel, A. C. (1987). *Expert systems: Strategic implications and applications.* New York: Halsted Press.

Davies, R. (1986). Expert systems and cataloguing: New wine in old bottles? In F. Gibb (Ed.), *Expert systems in libraries* (Proceedings of a conference of the Library Association Information Technology Group and the Library Association Research Group, November 1985) (pp. 67-82). London: Taylor Graham.

Fenly, C., & Harris, H. (1988). *Expert systems: Concepts and applications.* Washington, DC: Cataloging Distribution Service, Library of Congress.

Gibb, F., & Sharif, C. (1988). CATALYST: An expert assistant for cataloguing. *Program: Automated Library and Information Systems, 22*(1), 62-71.

Gorman, M., & Winkler, P. W. (1988). *Anglo-American cataloguing rules.* (2d rev. ed.). Chicago, IL: American Library Association.

Greene, R. (1990). *Implementing Japanese AI techniques: Turning the tables for a winning strategy.* New York: McGraw-Hill.

Hjerppe, R.; Olander, B.; & Marklund, K. (1985). *Project ESSCAPE: Expert systems for simple choice of access points for entries: Applications of artificial intelligence in cataloging.* Linköping, Sweden: Linköping University, Department of Computer and Information Science and University Library.

Hollnagel, E. (1989). The reliability of expert systems: An inquiry of the background. In E. Hollnagel (Ed.), *The reliability of expert systems* (pp. 14-36). Chichester, West Sussex, England: Ellis Horwood.

Kang, K. C., & Levy, L. S. (1989). Software methodology in the harsh light of economics. *Information and Software Technology, 31*(5), 239-250.

Olsen, P. R. (1989). Safety and risks in the use of financial expert systems. In E. Hollnagel (Ed.), *The reliability of expert systems* (pp. 119-133). Chichester, West Sussex, England: Ellis Horwood.

Prerau, D. S. (1985). Selection of an appropriate domain for an expert system. *AI Magazine, 6*(2), 26-30.

Prerau, D. S. (1990). *Developing and managing expert systems: Proven techniques for business and industry.* Reading, MA: Addison-Wesley.

Rolston, D. W. (1988). *Principles of artificial intelligence and expert systems development.* New York: McGraw-Hill.

Waterman, D. A. (1986). *A guide to expert systems.* Reading, MA: Addison-Wesley.

STUART WEIBEL

Research Scientist
OCLC Office of Research
Dublin, Ohio

Automated Cataloging: Implications for Libraries and Patrons

ABSTRACT

Many changes in cataloging practice have been and will increasingly be technology driven. Bound lists and drawers of cards defined the form and function of catalogs for as long as they existed until the advent of digital computers. Even today, however, MARC records are as much a derivative of catalog cards as the reverse. The additional functionality of computer catalogs affords opportunities to increase the effectiveness of the cataloging process and improve the value of the catalog itself. Three main research areas are examined with regard to their anticipated influence on this evolution. Automated cataloging research, focusing on the application of rule-based systems to cataloging, represents a novel way to address the cataloging process per se, but has as yet made only modest progress. The incremental implementation of a variety of computer-assisted methods for addressing aspects of cataloging represents a second, more conventional approach to advancing the state of the art in cataloging automation. This approach shares the goal of the first—to build intelligent capabilities into cataloging systems—but the focus remains on human cataloging systems and the methods of implementation are more conventional. The third area is not part of traditional concepts of cataloging at all, but will have a major impact upon what is available in catalogs in the broadest sense of that term. This "nontraditional" cataloging involves automated processing of documents to extract bibliographic information as well as full text. It will expand the range of cataloged objects to include items not generally

cataloged due to resource constraints. Automated processing of such materials will be characterized by lower quality and less complete cataloging, but will nonetheless promote improved access to materials that are currently lost to bibliographic control.

INTRODUCTION

Mark Twain is said to have remarked that when a writer is tempted to use the word "very," he should write "damn" instead, prompting the editor to excise the offending word, thereby improving the quality of the writing. We might apply Twain's advice to the phrase *artificial intelligence*. We are in fact concerned about making library processing more efficient and providing greater value to the patron. Whether this is under the guise of artificial intelligence (AI) or simply intelligent system design is of little consequence to a librarian or a patron. Some of the projects described here fall into the classical (if that is the proper word) AI category, but others are simply the well-considered application of human intelligence captured in the idiom of what once was described as intelligent programming. In aggregate, these projects promise more intelligent systems that ease the burden of catalogers, patrons, and perhaps even library budget officers.

There are many ways to characterize the procedures and results of cataloging processes, and the particular perspective one adopts will necessarily influence the characterization of progress and prospects. It is therefore useful to provide a perspective on the salient issues that will serve as a common foundation for further discussion.

The Cataloging Process: Two Perspectives

The library's perspective on cataloging is as a technical process that can be measured in terms of books processed, items added, shelf lists lengthened, and catalogers employed. The concern of the library is to efficiently and expeditiously provide access to the holdings of the library. Thus, cataloging activity is a bottleneck in making available newly acquired material to the patron and represents a major commitment of staff resources. The existence of OCLC, RLG, Utlas, and other technical service companies is ample testimony to the incentive to reduce the cost of cataloging through resource sharing. Automation of cataloging processes represents a further opportunity to reduce these costs and will therefore be a major concern of the library.

The patron's perspective on cataloging has to do with what she or he can or cannot find in the catalog. Electronic catalogs have made the searching and identification of materials more flexible and effective, and future improvements in online public access catalogs promise further benefits. To the extent that cataloging practice influences what is available and how it is located, the patron is influenced.

The Cataloged Object

The changing technology of today's library automation environment is having a major impact on the catalog. There is by no means complete agreement as to the desirability of this change, but there can be no argument that the change is ongoing and is having dramatic consequences. The traditional catalog was a collated list of monographs, periodicals, and a variety of special objects; this notion retains a vigorous existence in most of our minds. Increasingly, however, a catalog entry is a surrogate for an item that may exist at a location distant from the physical location of the catalog and is available either by loan or online. It is more often a *work* that is the object of our desire rather than the physical object itself.

Patrick Wilson (1989) captures this notion in his essay entitled "The Second Objective." Wilson proposes a rethinking of bibliographic organization that emphasizes the organization of *works* rather than a particular manifestation of a work and provides access to these works independent of their geographic location or current state of revision or reprinting.

The Catalog

The idea of the catalog itself is enlarged by advancing technological capability. John Duke (1989) proposes the notion of a virtual catalog with "tripartite record structure," a three-tiered catalog that encompasses everything but the physical artifact.

1. Document Surrogates: the traditional notion of an abbreviated, formalized citation structure.
2. Document Guides: synopsis of content—tables of contents, indexes, abstracts, summaries; an "enriched" record.
3. Document Texts: the work itself in electronic or digital format available for distribution.

It is this enlarged idea of cataloging (and the catalog itself) *a la* Wilson and Duke that will have a direct, substantive impact on the library and library patrons. In order to describe what this impact might be, it is useful to distinguish three distinct areas of research and the prospects for each.

AREAS OF CATALOGING AUTOMATION RESEARCH

These areas are complementary and overlapping. There are activities that easily fall into more than one of them, but they cover, in aggregate, the range of automation activities that have a strong influence on cataloging practice.

- fully automated cataloging; cataloging untouched by human hands;
- computer-assisted cataloging; tools or utilities, either active or passive, that could enhance the human cataloger's productivity; and
- nontraditional cataloging; automated processing of materials not typically included in conventional cataloging workflow.

Fully Automated Cataloging

The concept of a cataloging robot lies at the center of this area. The goal is to embed cataloging expertise in a system that has access to machine-readable versions of items to be cataloged and generate appropriate bibliographic surrogates and guides in an automated way. No one working toward this goal can long harbor illusions about the near-term prospects in this area. Nonetheless, the results of such work can have important side effects for real cataloging systems and can as well, perhaps, teach us something about what the successor to AACR2 should look like. Indeed, this last outcome is suggested by some to be the most important potential result.

The research environment supporting this area is a difficult one. Conceptual analysis is helpful, but at some point the proof is in the cataloging, and without convincing demonstrations of actual cataloging by machine, the effort is sterile. Building prototype systems is a difficult and costly activity with a number of seemingly intractable problems.

The first study of the feasibility of automating the cataloging process was in a dissertation written by Martha Fox at the University of Illinois at Urbana-Champaign. This study set out to "determine whether the human intellectual process of cataloging bibliographic materials could be simulated by automatic, namely, objective, non-intuitive, computer techniques" (1972, p. 3). One of Fox's conclusions merits mention in the context of current work in this area:

> Finally, if librarians are to consider a system in which automated cataloging is to play a part, it is essential that the intellectual structure of the existing cataloging process be reexamined in light of the capabilities and operations performed by machine. (pp. 304-05)

Davies and James (1984) published the first attempts at actually encoding some component of cataloging rules, and although their attempt was somewhat bogged down in the implementation aspects of building a system, Davies (1986) subsequently described many issues which anticipated later efforts in this area. Note that Davies is not an advocate of fully automated cataloging systems, but rather proposes that the rule-based system work interactively with a cataloger.

Helga Schwarz (1986) of the Deutsches Bibliothekinstitut proposed an approach to automating the extraction of bibliographic descriptors from title pages in a three-step process: (1) recognition of types of data, (2) recognition of the function of data, and (3) applying appropriate rules to formulate a cataloging record. Unlike Davies, she counts herself among the advocates of the cataloging robot, acknowledging that it may not be a reality in the near future.

Elaine Svenonius and Mavis Molto have two papers (1990, in press) that address issues in automated cataloging. The goal of these studies is to advance the theoretical underpinnings of automated cataloging as well as to provide pragmatic methods that could be incorporated into actual systems. Among the virtues of these studies is their foundation in real data. The authors randomly selected English language monographs from the UCLA stacks and systematically applied their ideas to the data. The results are useful heuristics that can be employed in practical systems that could be implemented today.

These studies emerged from previous work (Svenonius et al., 1986) in this group addressing conceptual issues in cataloging that bear on the rules supporting the choice of name-access points. These efforts in aggregate illustrate the close interaction of rule structures and the consequences these structures have for automated systems. The obvious question emerges: Should changing technology influence the way rules are structured or should the technology simply implement the rules?

Ling-Hwey Jeng (1986, 1988) has explored the potential for automating cataloging using title page information incorporating AACR2 into a structure suitable for application in an automated environment. Her recent work addresses the structure of AACR2 rules and the implications for implementation in an automated environment (1990).

Dissertation research at UCLA by Zorana Ercegovac, under the direction of Harold Borko (Borko & Ercegovac, 1989) approaches another aspect of cataloging: map cataloging. These investigators explored issues in the application of written and unwritten procedures for assigning map authorship. This study recognizes that necessary expertise in such tasks extends beyond that which is articulated in formal rule sets, and such considerations must inform any successful attempt at automating these processes. The technological impediments of automated map cataloging far outweigh those of monograph cataloging, mitigating against application of such ideas; however, the authors suggest that their work might profitably be applied to training of catalogers.

The Automated Title Page Cataloging Project (Weibel et al., 1989) at OCLC represents an attempt to demonstrate the feasibility of descriptive cataloging from title page images without the intervention of humans. The prototype was implemented as a rule-based system in Prolog; the objective of the system was to generate a first-level bibliographic description from information on the title page.

Sample title pages were selected randomly from current cataloging on the OCLC Online Union Catalog at the time of the study. Scanning and optical character recognition (OCR) were not (and are not now) sufficient to generate accurate representations of the title pages, so machine-readable versions of these title pages were rendered in a typesetting language and parsed automatically for the tests. In this way it was possible to tackle the conceptual problems associated with format recognition without being unduly handicapped by the realities of the technological limitations of scanning and OCR.

It is this thread of unreality that pervades to some extent all the automated cataloging studies alluded to above. They share an earnest attempt to address the conceptual problems in this area and a willingness to overlook the practical limitations that loom as large obstacles to implementation of production systems. This is not to say that such studies are fruitless; the value of these efforts lies in three areas:

1. providing a better understanding of the problems that must be solved to automate cataloging procedures,
2. pointing to productive ways to restructure cataloging procedures such that future automation attempts will have greater prospects for success, and
3. developing teaching simulators to enhance cataloging education.

They are unlikely to change technical processing in the library in the next five years, however.

Computer-Assisted Cataloging

Virtually all cataloging now performed in libraries is in some sense computer-assisted cataloging. For the purpose of this discussion, included somewhat arbitrarily are those tools not in common use but which will become more widespread in the near future. The implementation of such tools will have a major impact on technical processing departments. Are these artificial intelligence? Robert Burger (1984), in a paper entitled "Artificial Intelligence and Authority Control," made the statement: "artificial intelligence is already used in libraries...one of the major responsibilities of catalogers, machine-based authority control, is a form of artificial intelligence"(p. 344).

Whether such efforts should be considered artificially intelligent is moot; one may simply understand such capabilities as part of the naturally evolving capability of intelligent systems which support the cataloging effort. Several such projects in progress in the OCLC Office of Research illustrate the point.

Duplicate Detection

The Duplicate Detection Project at OCLC (O'Neill, 1989) is a good example of a practical implementation of a rule-based system that involves no specialized languages or unconventional techniques. It is the embodiment of a high degree of expert knowledge—the knowledge of an experienced cataloger—in combination with matching similarity algorithms to detect duplicate records in the OCLC Online Union Catalog. As such, it is a useful cataloging utility that has a variety of applications in a cataloging workstation as well as in its current batch processing implementation. A program such as this one which monitored cataloging input could contribute to preventing the addition of duplicate cataloging records rather than cleaning them up after the fact.

The current implementation has identified 80 percent of the duplicates in test samples with less than 0.5 percent misidentification of pairs of nonidentical records as duplicates.

Subject Heading Correction

A review article of online database quality control (O'Neill & Vizine-Goetz, 1988) describes a variety of error correction techniques that can be applied to databases. One of the authors, Edward T. O'Neill, currently is leading an effort to correct errors in subject headings in the OCLC Online Union Catalog. Two million records have been corrected in the initial phases of this effort; a million or more are expected to be corrected in a second phase. These efforts improve the quality of cataloging databases, thereby making cataloging more effective and making catalogs more useful to patrons.

Cataloger's Assistant

Diane Vizine-Goetz (1989) is leading a team that is developing a prototype cataloger's workstation for use in actual cataloging production. The system is now being tested at Carnegie-Mellon University Libraries for reclassifying a mathematics and computer science collection and applying subject cataloging to new items in these subject areas. The prototype makes available the Dewey Decimal Classification (DDC), machine-readable Library of Congress Subject Headings (LCSH-mr), and OCLC cataloging data in a HyperCard interface on the Apple Macintosh. The goal of this study is to explore the following issues in a production cataloging environment:

- How can the structure of DDC and LCSH best be conveyed to the user? How should these systems be linked?

- What browsing and navigational tools are appropriate for this application?
- What searching capabilities are necessary to support effective usage of these resources?

Nontraditional Cataloging

The term "nontraditional cataloging" is intended to describe the processing of materials that do not command the attention of a complete cataloging effort but should nonetheless be available for retrieval at some level, typically in an electronic database or catalog. The so-called "grey literature" or fugitive documents have traditionally fallen outside the body of fully cataloged items due to resource constraints or perceived lack of importance. Journal articles, pamphlets, correspondence, and office documents come to mind as examples of materials for which identification and retrieval are often substandard.

In addition, there are new forms of communication that are becoming widespread, such as E-mail and electronic newsgroups. Are such items worthy of cataloging? The answer to this question is a pragmatic one; they will be cataloged (in the broadest sense of the term) to the extent that the benefit is perceived to justify the cost. To the extent that cost is low and the process automated, more materials will be cataloged.

The goal is to capture in an automated way something like a cataloging record that is useful for search and retrieval. It is unlikely that automated systems will provide records of quality equivalent to human cataloging, but the increased access to an otherwise poorly accessible body of information should nonetheless be useful.

Project ADAPT

Project ADAPT is an ongoing project in the OCLC Office of Research to automate the conversion of paper documents to SGML-structured, machine-readable form and to provide searchable indexes for retrieving such documents.

The document representation continuum (see Figure 1) extends from the physical document (or its image) to a structured logical representation that includes the indexed text, associated graphics, and functional role of document objects (title, author, abstract, etc.), all represented in a database structure that will afford multiple views of the document and will support a wide variety of retrieval and presentation options.

The goal of Project ADAPT is to move incrementally from one end of the continuum toward the other. Image-based systems are now being produced for archiving and preservation activities. However, more sophisticated document representations can be expected to improve the utility and flexibility of such systems.

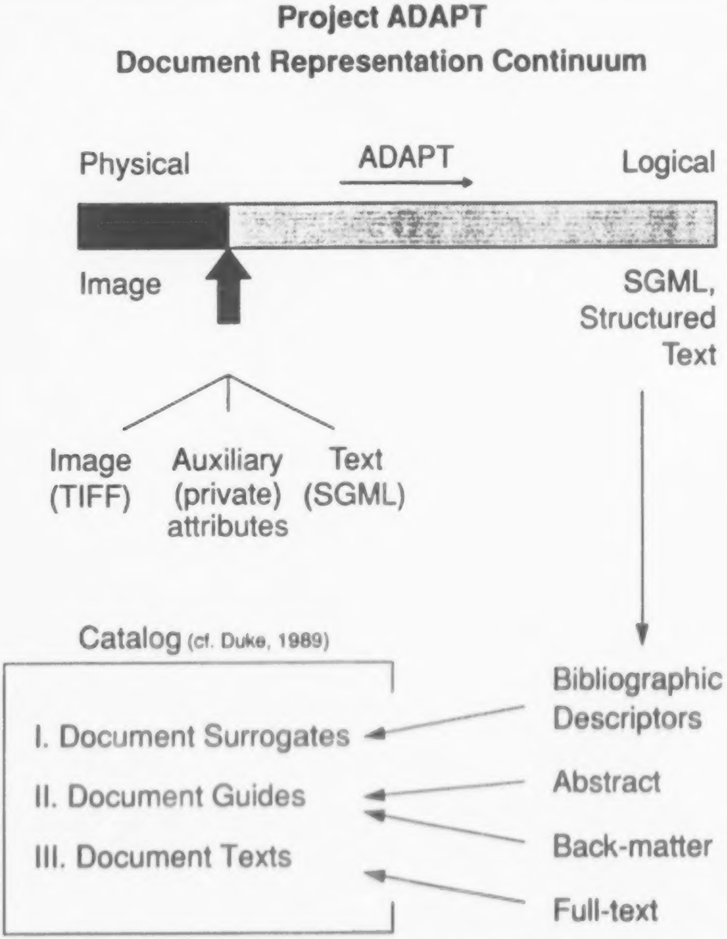

Project ADAPT
Document Representation Continuum
Physical
ADAPT
Logical
Image
SGML, Structured Text
Image (TIFF)
Auxiliary (private) attributes
Text (SGML)
Catalog (cf. Duke, 1989)
I. Document Surrogates
II. Document Guides
III. Document Texts
Bibliographic Descriptors
Abstract
Back-matter
Full-text

Figure 1. Document representation continuum

System Overview

Figure 2 represents the overall system design for a completed document processing system. The details of user interfaces and formatting of output are important production system considerations, but are of only minor concern to our project activities. The Newton Database server is also largely a production concern; it exists as the result of an intensive development activity in OCLC's development

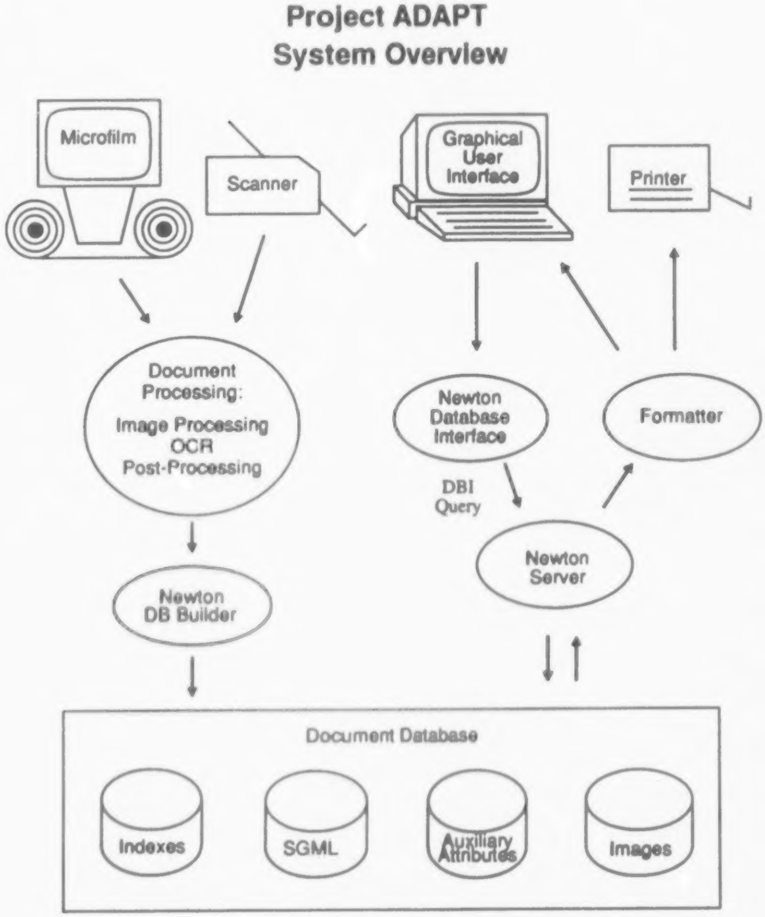

Project ADAPT
System Overview
Microfilm
Scanner
Graphical User Interface
Printer
Document Processing:
Image Processing
OCR
Post-Processing
Newton Database Interface
Formatter
DBI Query
Newton Server
Newton DB Builder
Document Database
Indexes
SGML
Auxiliary Attributes
Images

Figure 2. Project ADAPT system overview

department and is central to a number of OCLC products and research projects.

The overall strategy is to add value to commercial OCR capabilities by pre- and post-processing of the documents:

1. *Document Image Pre-processing:* The first processing stage is image pre-processing, which entails segmentation of the document image and classification of the segments into layout objects of several types. The segmentation process identifies rectangular layout objects for which a variety of statistical attributes are subsequently generated. These statistical attributes are then used to classify the objects as text, graphics, or extraneous noise. Text objects can then be passed through commercial optical character recognition devices.
2. *OCR Processing:* All OCR processing is done with commercial OCR systems. We use the Calera CDP 9000 system, but the process is designed to be independent of the OCR device used. Indeed, one of our strategies for reducing error in the processing is to employ multiple OCR processing and merge the results. This approach has resulted in reduction of errors by approximately 40 percent.
3. *Text Post-processing:* Post-processing includes activities from error correction to analysis of the structure of the document and markup of the document (in SGML) to reflect that structure. Document structure analysis involves the coordination of OCR output with associated layout objects such that the bitmap location of a particular line of text and other attributes of that line are accessible. This information can be used to determine the role of a particular text object (for example, titles, authors, and abstracts). This record, in conjunction with machine-generated indexes, then affords access to a work that might otherwise have little or no other means of retrieval.

Related Projects

There are many related projects in this country and elsewhere, each with somewhat different focus, but all with the common goal of making various types of written materials more readily available for search, retrieval, and distribution. Table 1 identifies representative projects. Not everyone will agree that the results of such processing will be uniformly good. Speaking about full-text access to cataloging records, Helen Schmierer (1989) writes: "Librarians and users will soon discover in online files of only moderate size that word access, while powerful, produces some bewildering results" (p. 112).

It is probably true that access to every word in a cataloging record or the document itself will raise many problems and much bewilderment, but the response should be to solve the problems rather than back away from them. In Duke's broad sense of the catalog, these systems are "cataloging" systems and they will ultimately promote greater access and availability of materials not now well represented in current catalogs. The occasional bewilderment and inherently greater complexity of retrieval in such a full-text world is a price that must be accommodated. Some of these problems will be mitigated by the maturation of the technology. In the long run, the patron will be well served by such capabilities.

TABLE 1
EXAMPLES OF TEXT DIGITIZATION PROJECTS

Organization	*Reference*	*Project Description*
Hochschule Darmstadt, Dept. of Information Science	Endres-Niggemeyer (1987)	AUTOCAT: OCR and automated cataloging of journal articles in the physical sciences
National Library of Medicine	Thoma et al. (1985)	Prototype system for electronic storage and retrieval of medical journal articles
OCLC Office of Research	Weibel et al. (1989)	Project ADAPT: automated document structure analysis and SGML markup
Nuclear Regulatory Commission	Bender (1988)	Optical disk-based system to deliver text, images search, and retrieval capabilities
National Agricultural Library	Andre & Eaton (1988), Zidar (1988)	Text Digitizing Project: scanning and OCR of agricultural documents for electronic retrieval and delivery
German National Research Center for Computer Science	W. Putz (personal communication, January 5, 1990)	Prototype system for conversion of paper documents to SGML structured documents
University of Strathclyde, Glasgow	F. Gibb (personal communication, January 24, 1990)	SIMPR: Software tools for indexing, retrieval, subject analysis, and structured information management

CONCLUSION: PROSPECTS FOR THE FUTURE

The projects described above are part of the foundation for future advances in the technology of cataloging. These projects provide a useful horizon to help gauge the future of cataloging, but the rate of progress toward such goals (and even the solidity of the goals themselves) is difficult to predict. Incremental progress will be made by implementing useful, practical cataloging tools—duplicate detection, more advanced authority checking, subject authority correction algorithms—in relatively conventional production environments.

The systems which result will be intelligently implemented rather than intelligent, they will be real rather than artificial, but, most importantly, they will make the cataloging process more practical and more efficient.

The cataloger will have increasingly sophisticated tools to augment the traditional process of cataloging, resulting in a better product at a lower cost. The patron will benefit from this by virtue of the indirect benefits of a more efficient operation.

As these parts of cataloging systems mature, research in the conceptual structure of cataloging and the automation of cataloging processes should provide a foundation to support longer term changes in cataloging and the systems to support it. When such changes take place, they will have also resulted from incremental progress toward an understandable goal. Some of the techniques that will have been applied to reach these goals are included in what are commonly understood to be artificial intelligence techniques; others will have been more conventional.

The other realm of potential improvements will come from the low end of the cataloging spectrum: the conversion of paper or microform to more accessible media—the electronic document. The large number of documents now in relatively unaccessible paper or microform that is not indexed or cataloged by humans can be rendered more accessible through a process of conversion to electronic format and machine indexing. The high level of research activity in this area suggests that systems to automate this conversion will have a major impact on cataloging information in the broadest sense of the word.

REFERENCES

Andre, P. Q. J., & Eaton, N. L. (1988). National agricultural text digitizing project. *Library Hi Tech, 6*(3), 61-66.

Bender, A. (1988). An optical disk-based information retrieval system. *Library Hi Tech, 6*(3), 81-85.

Borko, H., & Ercegovac, Z. (1989). Knowledge-based descriptive cataloging of cartographic publications. In *Annual review of OCLC research, July 1988-June 1989* (pp. 49-50). Dublin, OH: OCLC.

Burger, R. H. (1984). Artificial intelligence and authority control. *Library Resources & Technical Services, 28*(4), 337-345.

Davies, R. (1986). *Cataloguing as a domain for an expert system.* Chichester, England: Ellis Horwood.

Davies, R., & James, B. (1984). Towards an expert system for cataloguing: Some experiments based on AACR2. *Program: Automated Library and Information Systems, 18*(4), 283-297.

Duke, J. K. (1989). Access and automation: The catalog record in the age of automation. In E. Svenonius (Ed.), *The conceptual foundations of descriptive cataloging* (Papers presented at the Conference on the Conceptual Foundations of Descriptive Cataloging, 14-15 February 1987) (pp. 117-128). San Diego, CA: Academic Press.

Endres-Niggemeyer, B. (1987). Eine reprasentation der informationsstruktur von fachzeitschriften [A representation of the information structure of scientific journals (AutoCat)]. *Nachrichten fur Dokumentation, 38*, 333-340.

Fox, A. S. (1972). *The amenability of a cataloging process to simulation by automatic techniques.* Unpublished doctoral dissertation, University of Illinois at Urbana-Champaign, Illinois.

Ling-Hwey Jeng. (1986). An expert system for determining title proper in descriptive cataloging: A conceptual model. *Cataloging & Classification Quarterly, 7*(2), 55-70.

Ling-Hwey Jeng. (1988). The language of a title page. In C. L. Borgman & E. Y. H. Pai (Eds.), *ASIS '88: Proceedings of the 51st ASIS Annual Meeting* (23-27 October 1988) (pp. 31-35). Medford, NJ: Learned Information.

Ling-Hwey Jeng. (1990). *The general structure of a knowledge base for Anglo-American Cataloguing Rules: Final report* (Technical report of a project funded under the OCLC Library and Information Science Research Grant Program).

O'Neill, E. T. (1989). Duplicate detection. In *Annual review of OCLC research, July 1988-June 1989* (pp. 15-16). Dublin, OH: OCLC.

O'Neill, E. T., & Vizine-Goetz, D. (1988). Quality control in online databases. In M. E. Williams (Ed.), *Annual review of information science and technology* (Vol. 23, pp. 125-156). Amsterdam: Elsevier Science Publishers.

Schmierer, H. F. (1989). The impact of technology on cataloging rules. In E. Svenonius (Ed.), *The conceptual foundations of descriptive cataloging* (Papers presented at the conference on the Conceptual Foundations of Descriptive Cataloging, 14-15 February 1987) (pp. 101-116). San Diego, CA: Academic Press.

Schwarz, H. (1986). Expert systems and the future of cataloguing: A possible approach. *LIBER Bulletin, 26*, 23-50.

Svenonius, E.; Baughman, B.; & Molto, M. (1986). Title page sanctity? The distribution of access points in a sample of English language monographs. *Cataloging & Classification Quarterly, 6*(3), 3-21.

Svenonius, E., & Molto, M. (1990). Automatic derivation of name access points in cataloging. *Journal of the American Society for Information Science, 41*(4), 254-263.

Svenonius, E., & Molto, M. (in press). Studies in automatic cataloging. *Journal of the American Society for Information Science.*

Thoma, G. R.; Suthasinekul, S.; Walker, F. L.; Cookson, J.; & Rashidian, M. (1985). A prototype system for the electronic storage and retrieval of document images. *ACM Transactions on Office Information Systems, 3*(3), 279-291.

Vizine-Goetz, D. (1989). Cataloger's assistant. In *Annual review of OCLC research, July 1988-June 1989* (p. 8). Dublin, OH: OCLC.

Weibel, S.; Oskins, M.; & Vizine-Goetz, D. (1989). Automated title page cataloging: A feasibility study. *Information Processing & Management, 25*(2), 187-203.

Wilson, P. (1989). The second objective. In E. Svenonius (Ed.), *The conceptual foundations of descriptive cataloging* (Papers presented at the Conference on the Conceptual Foundations of Descriptive Cataloging, 14-15 February 1987) (pp. 5-16). San Diego, CA: Academic Press.

Zidar, J. A. (1988). National agricultural text digitizing project: System startup and operation. In M. E. Williams & T. H. Hogan (Eds.), *National Online Meeting Proceedings—1988* (Proceedings of the Ninth National Online Meeting, 10-12 May 1988) (pp. 443-448). Medford, NJ: Learned Information.

SUSANNE M. HUMPHREY

National Library of Medicine
Bethesda, Maryland

Interactive Knowledge-Based Systems for Improved Subject Analysis and Retrieval

ABSTRACT

A knowledge-based, or expert, system encoding both factual and procedural knowledge to assist users in performing an intellectual task is ideally suited to indexing. Existing thesauri, classification schemes, and indexing manuals are a good starting point, and artificial intelligence (AI) computer languages and data structures seem well suited for development of these systems. In addition, currently available workstation environments (with windows and mouse) and standard software (such as X Windows) should make possible sophisticated and portable interfaces. A unique prototype, the MedIndEx System, is being developed to assist people using the Medical Subject Headings (MeSH ®) thesaurus to index the MEDLINE ® database in MEDLARS ® (Medical Literature Analysis and Retrieval System) at the National Library of Medicine (NLM). MedIndEx is, in principle, applicable to any indexing system using a thesaurus and following a body of indexing rules.

INTRODUCTION

Human indexing of document collections for information retrieval entails assigning indexing terms to documents to facilitate their location by subject. These terms are selected from descriptors in thesauri, and after a document has been indexed, the document citation becomes part of a database, in which descriptors are a type of access point searchers may use for locating documents. Rules are established for guiding indexers in selecting these terms. These might be in the form of an indexing manual, notes in the thesaurus itself, specialized manuals from outside authorities adapted for the indexing system, collections of notes on complex topics that expand the principal manual, and even memoranda for emergencies that cannot await updates to other tools.

Since the expert system approach that will be discussed is derived from a unique prototype, the MedIndEx System, this paper will also reference the real-world indexing system that MedIndEx is designed to enhance. MedIndEx is designed to assist people using the MeSH thesaurus to index the MEDLINE database in MEDLARS at NLM. However, MedIndEx is, in principle, applicable to any indexing system using a thesaurus and following a body of indexing rules.

A knowledge-based, or expert, system, encoding both factual and procedural knowledge to assist users in performing an intellectual task, is ideally suited to indexing for MEDLINE. Facts, meaning concepts in the medical domain, have been recorded in MeSH over the years based primarily on literary warrant; thus the thesaurus has been developed specifically for representing the literature being indexed. Procedural knowledge would correspond to indexing rules. For these, there is the rather comprehensive *MEDLARS Indexing Manual,* supplemented for additional detail by specialized manuals (e.g., for tumors and enzymes), *Technical Notes,* and annotations appearing with individual descriptors in MeSH (hence the name *MeSH Annotated Alphabetic List* for the version used by indexers and searchers).

A special characteristic of MeSH that makes the conventional system particularly amenable to the expert system approach is the classification scheme, unifying all 15,000 descriptors in a single hierarchy known as *MeSH Tree Structures.* At a detailed level, MeSH trees are used for applying the *specificity* tenet of indexing, which states that a concept shall be indexed to the most specific term available; trees facilitate, and in fact define, this specificity by their hierarchical display. At a more general level, rules of coordinate indexing whereby a concept is expressed by assigning two or more descriptors are frequently stated as instructions to coordinate a term from one subclassification with a term from another. An example of this is the neoplasm coordination rule, which states that a neoplasm (cancer) is appropriately covered by a neoplasm-site term (from the *Neoplasms by Site* hierarchy, e.g., *Bone Neoplasms*) in coordination with a histologic-type term (from the *Neoplasms by Histologic Type* hierarchy, e.g., *Adenocarcinoma*). Furthermore, the *MEDLARS Indexing Manual* itself is largely organized by major MeSH categories (*Anatomy, Organisms, Disease, Drugs,* and so forth).

Another type of coordination between terms is characteristic of MEDLINE, whereby a descriptor is regarded as a main heading to which the indexer appends one or more subheadings as topical qualifiers. This is actually a form of *pre* coordination, since a main heading-subheading coordination is a semantic link created at indexing time. Although there are seventy-seven subheadings, not all of them may qualify all 15,000 descriptors. Rules for permissible main heading-subheading combinations again follow the MeSH classification scheme. For instance, with regard to the subheadings *CYTOLOGY* and *PATHOLOGY*, *Lung/CYTOLOGY*, *Lung/PATHOLOGY*, and *Lung Diseases/PATHOLOGY* are permitted, but *Lung Diseases/CYTOLOGY* is not. The general rule here is that terms in the *Anatomy* hierarchical category may be qualified by either subheading, but terms in *Disease* hierarchies only by *PATHOLOGY*. The basis for computational restrictions on permissible subheadings has recently been extended to hierarchies at the fourth level within a category; for instance, there is a set of restrictions on subheadings for the *Leukemia* hierarchy. Rules for main heading-subheading precoordination are conceptually compatible with the notion of inheritance, a most important feature of knowledge-based systems that will be mentioned again further on. However, inheritance of permissible subheadings is not performed computationally in the conventional system, as each descriptor record in MeSH contains explicit information about permissible subheadings.

In addition to the classification scheme, conventional indexing suggests relations that might be quite useful in an expert system. For instance, the BODY-SITE relation is often the basis for another type of precoordination, that is, a single indexing term encompassing two or more descriptors, such as *Lung Diseases (Disease* BODY-SITE *Lung)*, *Angiography (Radiography* BODY-SITE *Arteries)*, *Thoracic Surgery (Surgery* BODY-SITE *Thorax)*, *Gastritis (Inflammation* BODY-SITE *Stomach)*, and numerous others. The *Neoplasms by Site* hierarchy, mentioned earlier, consists mostly of precoordinate terms based on BODY-SITE.

The idea of coordination is, according to the *MEDLARS Indexing Manual*, "to give the clearest picture of the article within the limits of MeSH." This is related to the tenet of *multiplicity*, which is to provide for each document as many indexing terms as necessary to index it adequately from all its aspects—in other words, to index the document completely. There are many examples of coordination throughout the manual, for instance, "anticonvulsant therapy of epilepsy causing abnormalities" is indexed *Anticonvulsants/ADVERSE EFFECTS* + *Abnormalities, Drug-Induced* + *Epilepsy/DRUG THERAPY*. Note this includes precoordinations (single-term and main heading-subheading) as well as coordination of indexing terms. Relations that might be useful for assistance in achieving these sorts of coordination include ADVERSE-EFFECT (*Anticonvulsants*

ADVERSE-EFFECT *Abnormalities, Drug-Induced*), ETIOLOGY (*Abnormalities, Drug-Induced* ETIOLOGY *Anticonvulsants*), SUBSTANCE (*Drug Therapy* SUBSTANCE *Anticonvulsants*), PROBLEM (*Drug Therapy* PROBLEM *Epilepsy*), PROCEDURE (*Epilepsy* PROCEDURE *Drug Therapy; Anticonvulsants* PROCEDURE *Drug Therapy*).

As will be seen further on, these hierarchies and implied relations are important conceptually for developing the MedIndEx knowledge base to provide indexing assistance. However, it should be noted that precise hierarchies developed for conventional indexing are, to a significant degree, not directly usable, especially for inheritance, in that, understandably, principles have not been developed for this more rigorous computational use. In the conventional system, nonhierarchical relations do not exist as such in MeSH, and a set of relations has not yet been identified for computational use.

As an aside, admittedly an intermediate tack could be followed for indexing assistance, which is merely providing a workstation interface to improve interactive accessibility of the thesaurus. Two conditions make this feasible: MeSH is in machine-readable form, and conventional indexing is already performed in an interactive environment. In the mouse and windowing environment of workstations, entering a descriptor might automatically open a MeSH window on the screen, displaying this descriptor with its annotations, cross-references, immediate hierarchically related terms, and list of permissible subheadings. Indexing terms in this window might be mouse-able. Another window might be available for quick searching of the entire MeSH hierarchy. However, it would quickly become clear that this approach would be greatly enhanced by a knowledge base. For instance, computationally why repeat the same neoplasm coordination rule in the text of the annotation for every neoplasm term in MeSH, when, using hierarchies, the rule might be encoded only in a high-level neoplasm term and accessed by subordinate terms merely by virtue of their relationship to this term? Moreover, why display this rule unnecessarily if the indexer has already applied it? When occasionally "see related" cross-references are displayed with terms in this MeSH window, why are there not more of these, and why do they suggest related terms in some places when it would seem the same relationship holds in others and there is no cross-reference there?

Each of these questions suggests why encoding procedural knowledge is superior to displaying procedures as textual instructions. The first instance shows that hierarchies should be used computationally, to avoid redundant data and insure that rules for a class of concept are always applied to each member of that class. The second, that in addition to using relations, an expert system should capture and use knowledge of users' previous actions during system use. The third, that domain relations other than hierarchy, when applied systematically, would facilitate the indexing practice of giving the clearest picture of the article—multiplicity as mentioned earlier.

Finally, we bring up the question, why apply expert systems to indexing when the real problem is retrieval? Indexing variability and retrieval variability are well known in the field of information retrieval. Indexers indexing the same document using the same indexing system will often enough assign different sets of indexing terms; searchers searching the same database using the same retrieval system will retrieve different sets of citations for the same query. Indexing variability is in part responsible for retrieval variability; that is, as long as searchers need to compensate for indexing variability, their strategies for doing this will probably vary. If indexing were less variable, then searchers could rely increasingly on standard indexing practice. Furthermore, the same expert system used by indexers, facilitating consistency with an expert standard, would be adaptable for searchers as well; on the other hand, without expert indexing, there is less chance of developing precise expert systems for searchers. Therefore, an expert system for indexing, all other things being equal, would improve retrieval by serving to remove at least one of the causes of retrieval variability.

Figure 1 is a diagram of the MedIndEx System showing four main processes: Journal Assignment, Indexer Interface, MeSH Indices Report Generator, and MedKB Manager (MedKB is the name of the knowledge base). Also known as P1 - P4, respectively, they are summarized as follows:

- P1—Journal Assignment Utility. The main purpose of the Journal Assignment Utility program is to control assignment of Journal Source Files (consisting of bibliographic citations without indexing terms) to indexers. This program must be run before an indexer may use P2. The prototype version of this program will allow the project officer to assign a journal issue to multiple indexers in order to conduct experiments. However, in an operational environment the system will protect against multiple assignment.
- P2—Indexer Interface. The Indexer Interface program provides indexers with a guided data entry indexing session for documents to be indexed. Journal issue assignments are made available to the indexer using P1, and the system prompts for MeSH indexing terms for assigned documents. MedKB and Word File, created by P4, are needed for providing corresponding indexing frames and assisting indexers in filling them in with additional indexing terms. The user may redo any indexing frame as long as the document is being indexed. When indexing of a document is completed, the instantiated document frame, with its indexing frames that have been created, becomes part of the database of indexed citations. Also stored in

the database are conventional MEDLINE indexing terms generated by the program using indexing frames and MedKB. These terms are used for evaluation by P3.

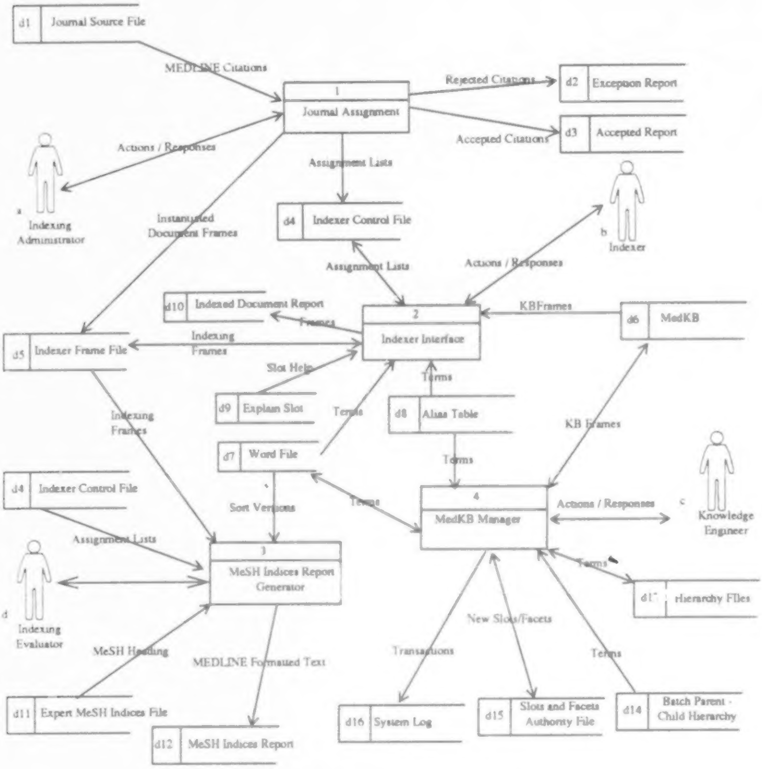

d1 Journal Source File
MEDLINE Citations
1
Journal Assignment
Rejected Citations
d2 Exception Report
Accepted Citations
d3 Accepted Report
Actions / Responses
a
Indexing Administrator
Assignment Lists
Instantiated Document Frames
d4 Indexer Control File
Assignment Lists
Actions / Responses
b
Indexer
d10 Indexed Document Report
Frames
2
Indexer Interface
KBFrames
d6 MedKB
d5 Indexer Frame File
Indexing Frames
Slot Help
Terms
d9 Explain Slot
Terms
d8 Alias Table
KB Frames
Indexing Frames
d7 Word File
Terms
d4 Indexer Control File
Sort Versions
Terms
4
MedKB Manager
Actions / Responses
c
Knowledge Engineer
Assignment Lists
3
MeSH Indices Report Generator
d
Indexing Evaluator
Terms
d17 Hierarchy Files
New Slots/Facets
MeSH Heading
MEDLINE Formatted Text
Transactions
Terms
d11 Expert MeSH Indices File
d12 MeSH Indices Report
d16 System Log
d15 Slots and Facets Authority File
d14 Batch Parent - Child Hierarchy

Figure 1. MedIndEx prototype data flow diagram

- P3—MeSH Indices Report Generator. The MeSH Indices Report Generator program produces a report to assist the project officer in evaluating information entered by indexers during P2, in terms of conformity to an expert standard. MeSH indices (conventional MEDLINE indexing terms produced by using P2) may be compared with each other, or with expert MeSH indices (specially certified by NLM's Index Section for this project) to determine the consistency of the system. Numeric scores are calculated based on Hooper's Indexing Consistency Measurement.

- P4—MedKB Manager. MedKB Manager is a knowledge base management system (KBMS) used by knowledge engineers to create and edit frames, which are data structures in which MedKB is written. P4 also is used for creating new entries (official terms, aliases, and sort versions) to be written into a Word File, including official terms, aliases, and sort versions. MedKB and Word File are essential to using P2.

Further details will appear in the system design section of the third NTIS report on MedIndEx (Humphrey & Chien, 1990). The current paper will discuss use of the Indexer Interface and MedKB Manager, respectively. Other reports on MedIndEx have been published giving additional background and references (Humphrey, 1989a, 1989b; Humphrey & Miller, 1987).

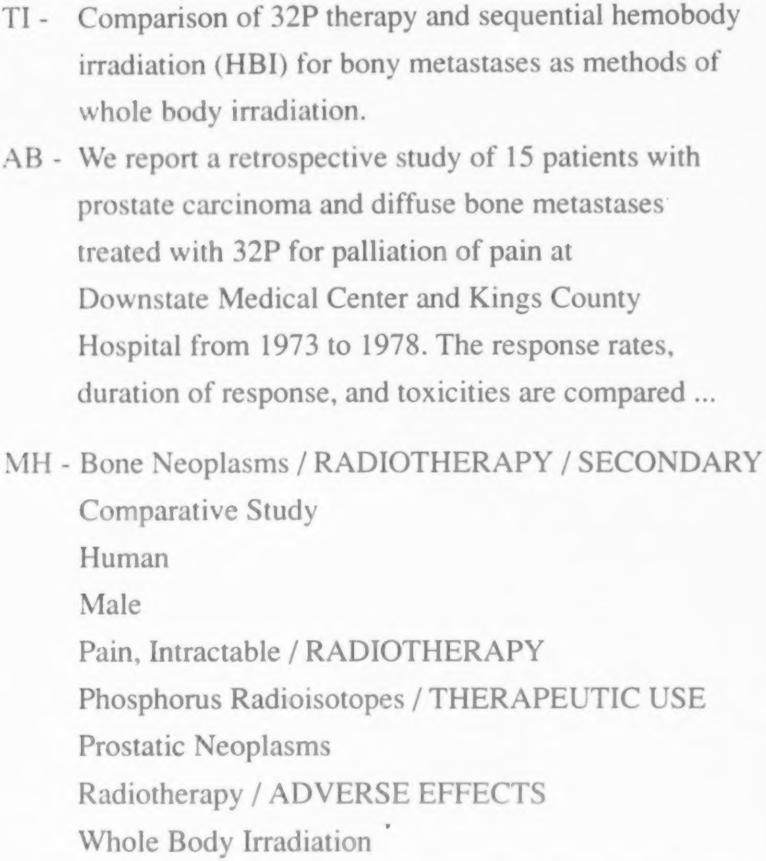
KNOWLEDGE-BASED INDEXING

This section will describe indexing using the MedIndEx Indexer Interface, specifically:

- The data structure that is used, consisting of frames, subdivided by slots, subdivided by facets.
- Knowledge-based assistance encoded in facets (object-oriented approach).
- The inheritance mechanism for accessing procedures and data, whereby *indexing frames* inherit from *knowledge-base frames*, knowledge-base frames inherit from each other.
- Internal retrieval for accessing and displaying data from other locations.

As a brief reminder of conventional indexing, indexers combine reading and scanning full-text journal articles (also referred to as documents in this paper) and select and assign terms from MeSH that best describe what these articles are about. Figure 2 shows the title, abstract, and indexing terms, labeled *MH* (for *MeSH Heading*), for an article about certain types of radiotherapy to treat pain of bone cancer. Indexing using MedIndEx is basically the same as conventional indexing regarding use of MeSH and adherence to tenets of indexing (e.g., specificity and multiplicity, as discussed earlier). Users are presumed to be trained MEDLINE indexers. However, unlike conventional indexing, MedIndEx provides interactive, situation-specific assistance in applying indexing rules. This is accomplished by filling in indexing frames with MeSH terms rather than merely listing these terms as individual occurrences. A frame is a data structure consisting of a frame name, like Bone Neoplasms (MeSH term for "bone cancer"), and slots which label conceptual relations linking the current frame to other frames.

Selecting *Bone Neoplasms* as the initial indexing term causes the system to display the first indexing frame (Figure 3). The frame name



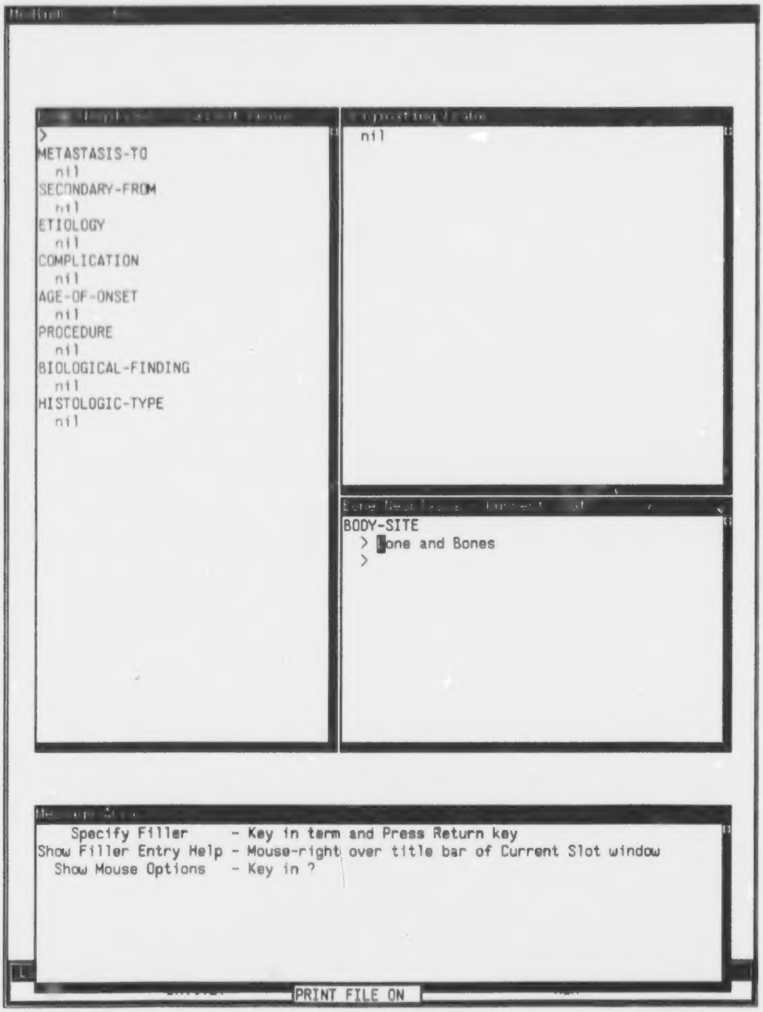
TI - Comparison of 32P therapy and sequential hemobody irradiation (HBI) for bony metastases as methods of whole body irradiation.

AB - We report a retrospective study of 15 patients with prostate carcinoma and diffuse bone metastases treated with 32P for palliation of pain at Downstate Medical Center and Kings County Hospital from 1973 to 1978. The response rates, duration of response, and toxicities are compared ...

MH - Bone Neoplasms / RADIOTHERAPY / SECONDARY
Comparative Study
Human
Male
Pain, Intractable / RADIOTHERAPY
Phosphorus Radioisotopes / THERAPEUTIC USE
Prostatic Neoplasms
Radiotherapy / ADVERSE EFFECTS
Whole Body Irradiation

Figure 2. Title, abstract, & MeSH indexing terms for radiotherapy article

Bone Neoplasms is in the title bar of the Current Frame and Current Slot windows. A Message Area window displays system messages. Slots in the Current Frame window are METASTASIS-TO, ETIOLOGY, PROCEDURE, and so forth. These are names of relations relative to the frame term, and will be used as prompts for filling in this frame.



```
>
METASTASIS-TO
  nil
SECONDARY-FROM
  nil
ETIOLOGY
  nil
COMPLICATION
  nil
AGE-OF-ONSET
  nil
PROCEDURE
  nil
BIOLOGICAL-FINDING
  nil
HISTOLOGIC-TYPE
  nil
```

```
  nil
```

```
BODY-SITE
  > Bone and Bones
  >
```

```
    Specify Filler      - Key in term and Press Return key
Show Filler Entry Help - Mouse-right over title bar of Current Slot window
  Show Mouse Options    - Key in ?
```

Figure 3. Sample Bone Neoplasms indexing frame on (MedIndEx)

The current slot, BODY-SITE, is displayed in the Current Slot window. Prompted by the greater-than sign, indexers may enter as slot fillers other frame names that complete the relationship "frame—slot—filler." For instance, indexer verification of the filler in this window

Bone and Bones would complete the relationship *Bone Neoplasms—BODY-SITE—Bone and Bones,* indicating that the article is about bone cancer of bones in general. The Current Frame window contains the remaining slots, which have not yet been processed (*nil* indicates absence of fillers). When the indexer is finished filling the current slot, BODY-SITE, it will be returned to its marked position in the Current Frame window, with its fillers, and the next slot, METASTASIS-TO, will take its place in the Current Slot window. This is how all slots are eventually processed.

There are two things of particular significance for understanding how the system works that are *not* shown in the foregoing slides of indexing frames:

- First, the fact that the actual name of this indexing frame is *Bone Neoplasms 86265451.* Internally, all indexing frame names consist of two parts: a MeSH term + the unique identifier of the article being indexed.
- Second, not shown is the *relationship* between this indexing frame and its corresponding MedKB frame.

If this information were shown in the Current Frame window, it would appear as follows:

Bone Neoplasms 86265451-Current Frame
INHERITS-FROM
>[Bone Neoplasms]

Frames in MedKB do more than provide terminologic authority control, as does a thesaurus. MedKB frames, such as the *Bone Neoplasms* frame (Figure 4), are subdivided by slots with associated information. Most of these slots represent domain-specific relations not found in conventional thesauri, which typically exhibit only synonymy, broader/narrower term relations, and general relatedness. Note these slots are the same as those in the indexing frame (Figure 3), and in fact they came from this MedKB frame. The *Bone Neoplasms* MedKB frame can be thought of as a template encoding data and procedures for interactively assisting indexers in filling slots of indexing frames like *Bone Neoplasms 86265451.* The code would be where the dots are, following each slot name. During the indexing session, indexing frames are constantly accessing their template MedKB frame via the INHERITS-FROM slot. This sort of accessing exemplifies the mechanism known as *inheritance.* In fact, when an indexing frame is first created—a procedure known as *instantiation*—it does not explicitly contain any of the slots presented on the screen (Figure 3), but merely accesses them, or inherits them, from its corresponding MedKB frame, and a screen management program causes them to be displayed. Only when a slot first appears in the Current Slot window is it physically created in the indexing frame.

To begin a few examples of MedIndEx assistance, use of slots to prompt for indexing terms is considered a form of assistance to help ensure more complete indexing, as it focuses indexers' attention on aspects of topics that should be considered. In Figure 3 the *Bone and Bones* filler for BODY-SITE was displayed automatically as a default filler for this slot, which an indexer may erase or over-write.

```
(|Bone Neoplasms|
  (INHERITS-FROM
    (VALUE |Bone Diseases| |Neoplasms by Site|))
  (CHILDREN ...)
; domain specific slots follow:
  (BODY-SITE ...)
  (METASTASIS-TO ...)
  (SECONDARY-FROM ...)
  (ETIOLOGY ...)
  (COMPLICATION ...)
  (AGE-OF-ONSET ...)
  (PROCEDURE ...)
  (BIOLOGICAL-FINDING ...)
  (HISTOLOGIC-TYPE ...))
```

Figure 4. Slots in MedKB frames

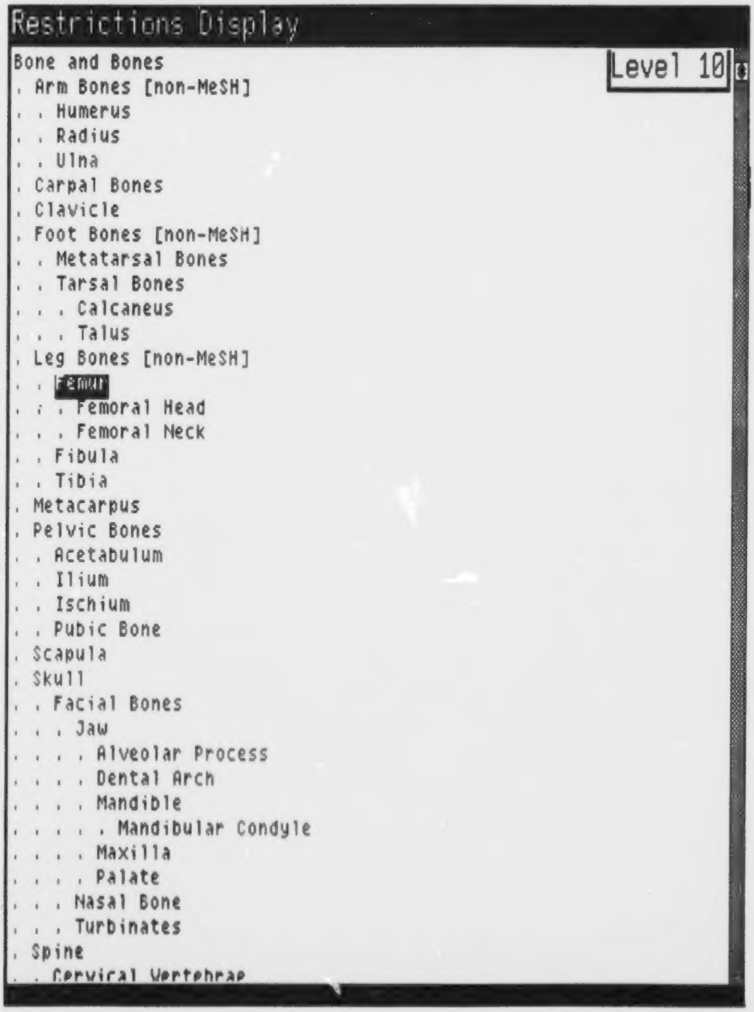
From the Current Slot window, indexers may request a display of permissible fillers for the current slot. These are considered "restrictions on the slot," and are listed in a Restrictions Display window, as shown in Figure 5, where the indexer has moused on a selection—namely, the particular bone *Femur,* intending for it to replace the default *Bone and Bones* (Figure 3). (The term *Femur* could also have been keyed in.)

However, in response to this entry (Figure 6) the system advised the indexer, in the Message Area window, that *Femur*—although sensible for this slot according to medical knowledge—is not permitted, since there is a more specific frame in the system, namely *Femoral Neoplasms*. In addition to this message, the system automatically erased *Femur* and restored the default *Bone and Bones* filler. This sort of help, which we call the *specificity* feature, is performed algorithmically. Accordingly, the system checked for child frames of *Bone Neoplasms* in MedKB. When it found one, *Femoral Neoplasms,* it checked if the current filler *Femur* is permitted for the current slot in this child frame. If so, then the indexing frame should be accessing this more specific child frame, *Femoral Neoplasms,* rather than *Bone Neoplasms.* For this sort of assistance a knowledge-based approach is essential. Alternatively, the *Bone and Bones* default filler might merely be erased, leaving the slot empty, but since a filler is required for this slot, the system would prevent exiting the slot if it remained empty.

Not all notification of more specific frames is of the type we have just seen, which is *prescriptive* (or enforced). Figure 7 shows *Bone Neoplasms* as a filler in the ETIOLOGY slot of a *Pain* frame, to which the system responded by merely *suggesting* that the more specific frame *Pain, Intractable* be considered. Here of course the system does not erase the indexer's entry, as it did *Femur* in the earlier example. We call this feature *relaxed specificity.* Similar assistance is provided when filling in the COMPLICATION slot in the *Bone Neoplasms* frame with the term *Pain,* where the system suggests the more specific term as filler.

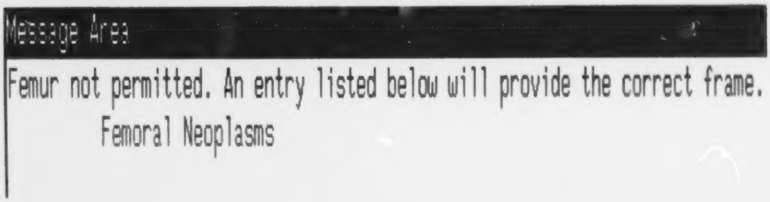
In Figure 8 the system displays a reminder to coordinate the fillers shown, *Radioisotope Therapy* and */DRUG THERAPY,* with a third filler *Combined Modality Therapy* if appropriate. The system recognizes existing fillers as representing different modalities, of radiation and drugs. The rule inherent in this reminder is, in case two or more treatments of different modalities are combined in treating the same patients, *Combined Modality Therapy* should be entered as well.

Finally, there is the type of assistance where the system automatically displays certain fillers based on previous fillers, often in previously filled frames that have been stored. To continue after a frame is completed and stored, indexers request pop-up menus for selecting another frame for processing. One menu consists of names of stored frames, leading to editing these frames. The other consists of *new* frame terms, that is stored-frame fillers that have not yet been instantiated as frames themselves. After filling and storing the initial frame for the first time, there is only a list of new frame terms corresponding to fillers of this just completed frame. A document is ultimately represented by a network of filled, stored indexing frames. Using an ORIGINATING-FRAME slot in indexing frames, the system maintains links between the current frame and stored frames it came from, where it was a slot filler; information in these previous frames can be used for prescribing and suggesting fillers in the current frame, as illustrated by the following example.



```
Bone and Bones
. Arm Bones [non-MeSH]
. . Humerus
. . Radius
. . Ulna
. Carpal Bones
. Clavicle
. Foot Bones [non-MeSH]
. . Metatarsal Bones
. . Tarsal Bones
. . . Calcaneus
. . . Talus
. Leg Bones [non-MeSH]
. . Femur
. . . Femoral Head
. . . Femoral Neck
. . Fibula
. . Tibia
. Metacarpus
. Pelvic Bones
. . Acetabulum
. . Ilium
. . Ischium
. . Pubic Bone
. Scapula
. Skull
. . Facial Bones
. . . Jaw
. . . . Alveolar Process
. . . . Dental Arch
. . . . Mandible
. . . . . Mandibular Condyle
. . . . Maxilla
. . . . Palate
. . . Nasal Bone
. . . Turbinates
. Spine
. . Cervical Vertebrae
```

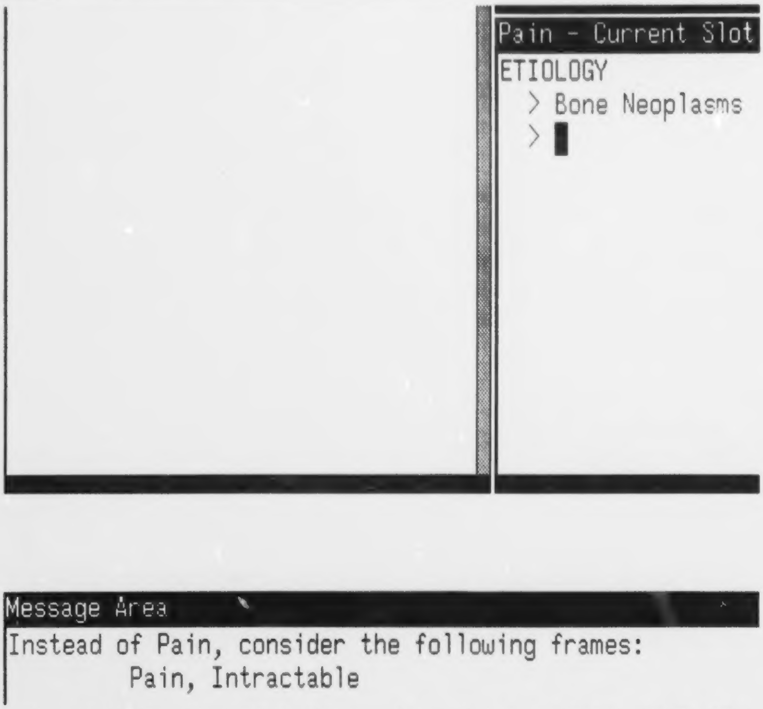
Figure 5. Sample restrictions display window (MedIndEx)



Message Area

Femur not permitted. An entry listed below will provide the correct frame.
        Femoral Neoplasms

Figure 6. "Specificity" feature in message area window (MedIndEx)



Pain - Current Slot
ETIOLOGY
  > Bone Neoplasms
  >

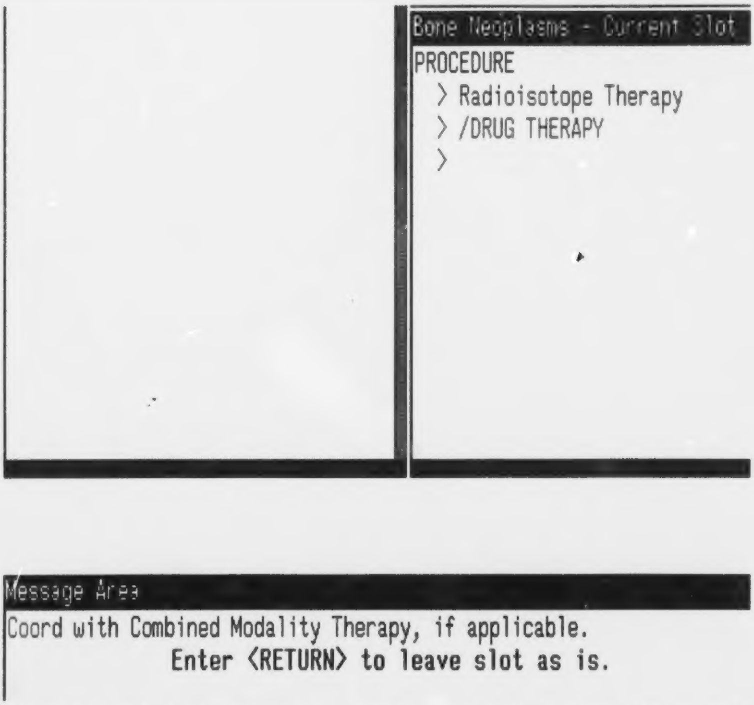
Message Area

Instead of Pain, consider the following frames:
        Pain, Intractable

Figure 7. "Relaxed specificity" feature in message area window (MedIndEx)

Figure 9 shows the next frame selection *Radioisotope Therapy,* where **PROBLEM** is the first current slot. The expression in the Originating Frame window, *Bone Neoplasms PROCEDURE [A],* is interpreted as follows: the current frame name ***Radioisotope Therapy*** was an *[A]*dded filler in the PROCEDURE slot in the originating frame *Bone Neoplasms*. The indexer may mouse on the originating frame name to view this frame in a Display window overlaying the Originating Frame window.

The boldface type of ***Bone Neoplasms*** in the PROBLEM slot in the Current Slot window indicates that it cannot be erased; it was retrieved by the system internally and merely displayed in this slot based on information in the stored *Bone Neoplasms* indexing frame. The system has deduced that since, according to this stored frame, radioisotope therapy was the procedure for bone neoplasms, then bone neoplasms *must* be the problem for radioisotope therapy, and the frame term of the stored frame should be retrieved and displayed. Because of this reciprocity between PROCEDURE and PROBLEM, the *Bone Neoplasms* filler must be correct if the *Radioisotope Therapy* filler was correct in the stored frame.



```
PROCEDURE
  > Radioisotope Therapy
  > /DRUG THERAPY
  >
```

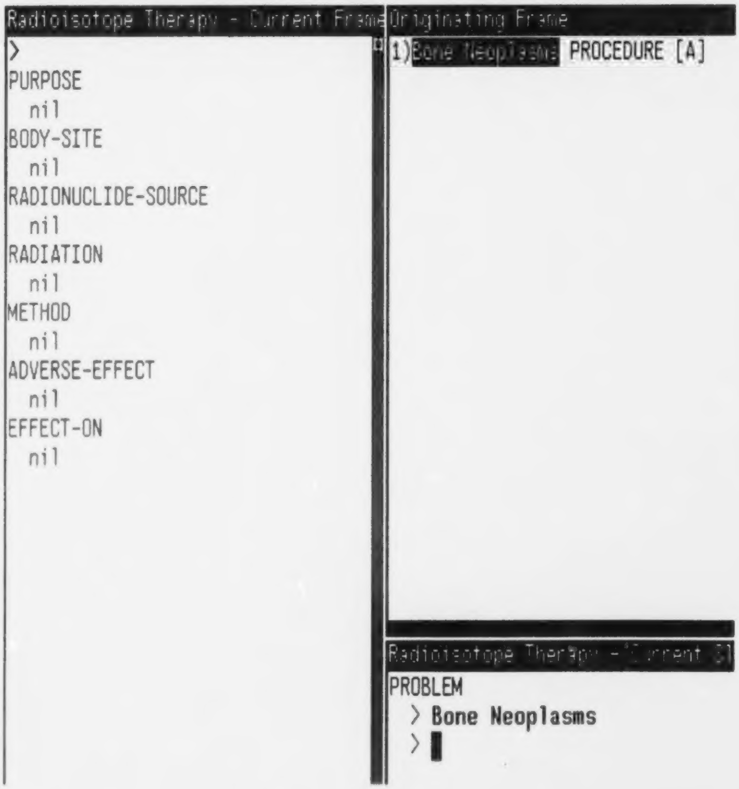
```
Coord with Combined Modality Therapy, if applicable.
              Enter <RETURN> to leave slot as is.
```

Figure 8. Reminder in message area window to coordinate fillers (MedIndEx)

We have presented a sample of available indexing assistance. In general, assistance may be categorized in several ways:

- Specificity and enforceability.
  - —Nonspecific suggestive (e.g., slot names as prompts, restrictions display).
  - —Specific suggestive (e.g., fillers displayed automatically in slots, fillers in messages).
  - —Nonspecific prescriptive (e.g., fillers in messages).
  - —Specific prescriptive (e.g., retrieved values, inherited values, specific required fillers).
- Data dependency.
  - —Authority dependency.
    - —MedKB (e.g., specificity of filler, passing restrictions, default filler, inherited values).
    - —Word File (e.g., checking if filler is an official term).
    - —Alias Table (e.g., checking if a word is acceptable).
  - —Other filler dependency (e.g., filler as coordinate with previous filler, superfluous filler with respect to previous filler).
- Procedural dependency.
  - —Automatically.
    - —Entering current slot (e.g., display of system-provided fillers).
    - —Processing a slot (e.g., checking if action is ok).
      - —Adding or verifying fillers.
      - —Erasing fillers.
      - —Attempting to exit a slot.
    - —Finishing a document (e.g., final consistency check).
  - —On request (e.g., restrictions/hierarchy display, consistency check).

The foregoing examples of assistance—and others—are encoded in structures known as *facets*. Facets are sublists within slots in MedKB frames. Figure 10 shows several types of facet, identified by labels SPECIFICITY, RESTRICTIONS, IF-NEEDED, DEFAULT, CAN-CONTINUE?, CAN-ADD?, subdividing some slots; code would be where the dots are. Figure 11, zooming in on the BODY-SITE slot in *Bone Neoplasms,* and showing actual facet code corresponding to some of the assistance described previously, is merely to illustrate briefly the importance of facets and how they work. Three facets are shown for this slot: CAN-CONTINUE?, DEFAULT, and IF-NEEDED. Explanations are in terms of effect of facets on indexing frames inherited from this MedKB frame, and thereby inheriting these facets. The *COND* statement in CAN-CONTINUE? checks for cardinality of 1, and if there is no filler in this slot in a *Bone Neoplasms* indexing frame, it sets the message that a filler is required and that the specific default is available. It gets the default *Bone and Bones* from the DEFAULT facet. The *SETQ* statement in the IF-NEEDED facet also gets this default, and provides it as a suggested filler when the slot is first presented in the Current Slot window.



```
Radioisotope Therapy - Current Frame
>
PURPOSE
  nil
BODY-SITE
  nil
RADIONUCLIDE-SOURCE
  nil
RADIATION
  nil
METHOD
  nil
ADVERSE-EFFECT
  nil
EFFECT-ON
  nil
```

```
Originating Frame
1)Bone Neoplasms PROCEDURE [A]
```

```
Radioisotope Therapy - Current
PROBLEM
  > Bone Neoplasms
  >
```

Figure 9. Sample originating frame/current frame window (MedIndEx)

A feature of MedIndEx is that it uses indexing frames to generate conventional MEDLINE indexing. There are two reasons why this is important. First, if the system were adopted, it would then generate the same sort of MeSH indexing output now used for MEDLINE. We still need this output for retrieval, since there is no language for retrieving against a frame database. Second, this output would facilitate testing the system by comparison with output from the conventional indexing activity.

```
(|Bone Neoplasms|
  (INHERITS-FROM
     (VALUE |Bone Diseases| |Neoplasms by Site|))
  (CHILDREN
     (SPECIFICITY ...))
  (BODY-SITE
     (RESTRICTIONS ...)
     (SPECIFICITY ...)
     (CAN-CONTINUE? ...)
     (DEFAULT ...)
     (IF-NEEDED ...))
  (METASTASIS-TO)
  (SECONDARY-FROM)
  (ETIOLOGY)
  (COMPLICATION
  (AGE-OF-ONSET)
  (PROCEDURE
     (CAN-ADD? ...))
  (BIOLOGICAL-FINDING)
  (HISTOLOGIC-TYPE))
```

Figure 10. Types of facet in MedKB frames

To illustrate, Figure 12 shows a *Radioisotope Therapy* indexing frame at a point where the filler *Phosphorus Radioisotopes* has been entered for the RADIONUCLIDE-SOURCE slot. The MeSH Indices

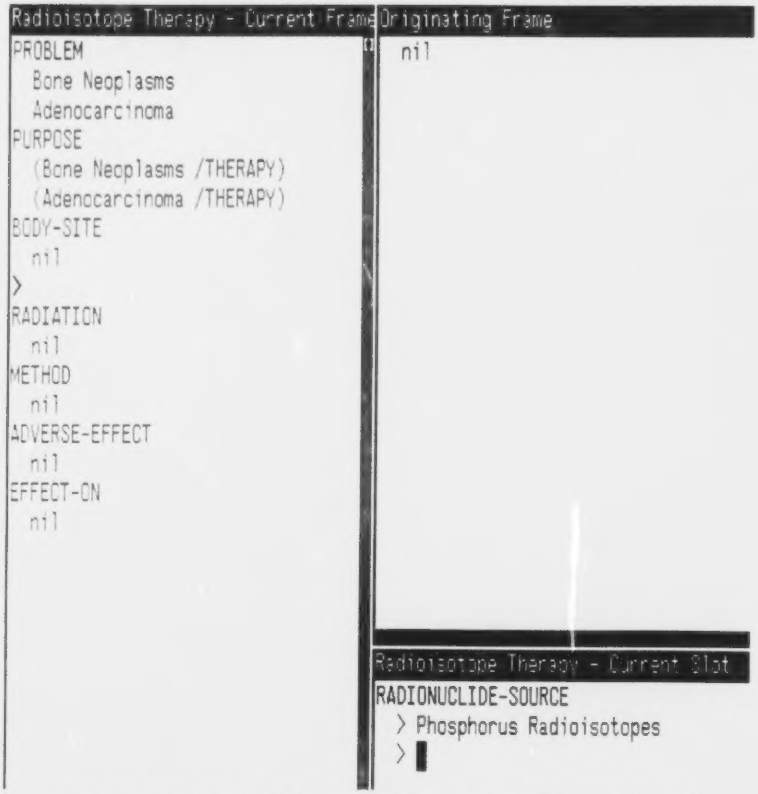
Display window (Figure 13) contains conventional, system-generated MEDLINE indexing, labeled *MH* (for *MeSH Heading*). Two precoordinated indexing terms have been generated so far: *Adenocarcinoma* and *Bone Neoplasms* appended by the same subheading *RADIOTHERAPY.* This display is updated automatically, in the background, as frame-filling proceeds for the remainder of this document. In this example, when the indexer exits the RADIONUCLIDE-SOURCE slot, thereby actually adding *Phosphorus Radioisotopes* to this slot in the indexing frame, and again requests this MeSH Indices Display, the system will show an additional MH entry, namely *Phosphorus Radioisotopes* with the subheading */THERAPEUTIC USE* (Figure 14). As illustrated by this example, the system is designed to perform all subheading assignment, thereby forming all main heading-subheading precoordinations, autonomously. Rules for automatic updating of MH entries are encoded in *IF-ADDED* facets attached to slots of MedKB frames.

```
(|Bone Neoplasms|
  (BODY-SITE
     (CAN-CONTINUE?
        (COND ((AND (NULL !FILLERS)
                         (SETQ DEF
                                 (GET-FILLERS !FRAME !SLOT 'DEFAULT)))
                  (SETQ !MESSAGE
                         (FORMAT NIL
"A filler is required for this slot. The following default is available: ~%"{~S;~}"
                                 DEF)) T))
     (DEFAULT |Bone and Bones|)
     (IF-NEEDED
        (SETQ  !RETRIEVED NIL
                !CANDIDATE
                         (GET-FILLERS !FRAME !SLOT 'DEFAULT))))))
```

Figure 11. Sample of actual facet code in MedKB frame

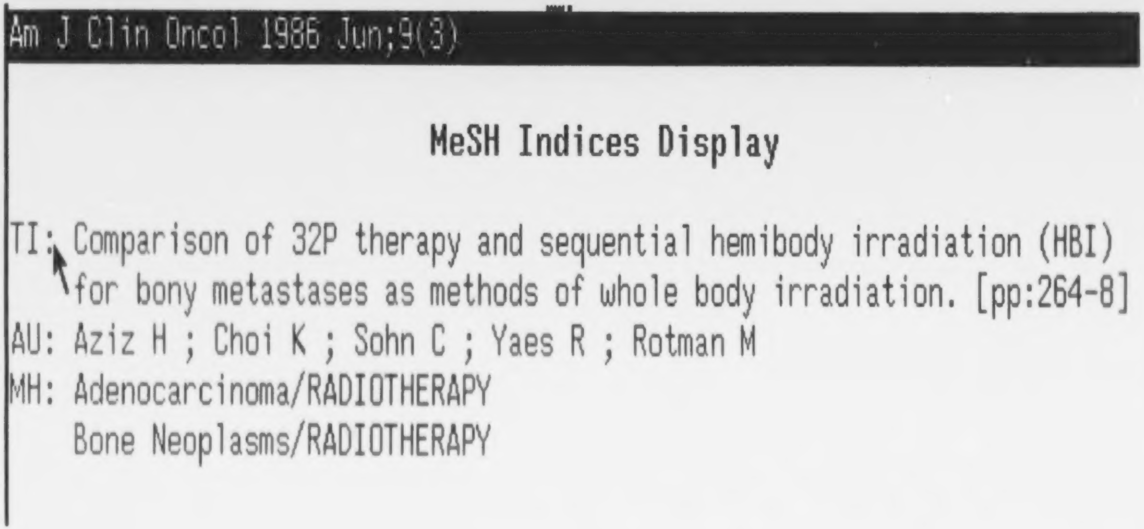
To simply give an idea of the subdivision of MedKB frames into slots and facets, computerized assistance for filling in the *Bone Neoplasms* indexing frame has been depicted as if encoded in the *Bone Neoplasms* MedKB frame, which all *Bone Neoplasms* indexing frames (such as *Bone Neoplasms 86265451*) inherit from. However, encoding these procedures, many of which would apply to all neoplasms, or even all diseases, in relatively low-level frames (such as *Bone Neoplasms*) would result in much redundancy in the knowledge base. In theory, a fillable slot can have as many as eight facets; a frame with eight fillable slots would therefore have sixty-four facets. If facets were explicit in each frame of MedKB, assuming about 15,000 frames (the approximate number of terms in MeSH), this would clearly make for an impossibly large knowledge base, with a great deal of redundancy.

Therefore, when we stated earlier that an indexing frame inherits data and procedures from the MedKB frame to which it bears the INHERITS-FROM relationship, this was not quite accurate. The MedKB frame from which it inherits might not explicitly encode the needed information either, and might, in turn, inherit it from another MedKB frame. Figure 15 shows the INHERITS-FROM ancestry of



```
Radioisotope Therapy - Current Frame   Originating Frame
PROBLEM                                  nil
  Bone Neoplasms
  Adenocarcinoma
PURPOSE
  (Bone Neoplasms /THERAPY)
  (Adenocarcinoma /THERAPY)
BODY-SITE
  nil
>
RADIATION
  nil
METHOD
  nil
ADVERSE-EFFECT
  nil
EFFECT-ON
  nil
                                       Radioisotope Therapy - Current Slot
                                       RADIONUCLIDE-SOURCE
                                         > Phosphorus Radioisotopes
                                         >
```

Figure 12. Sample Radioisotope Therapy indexing frame (MedKB)

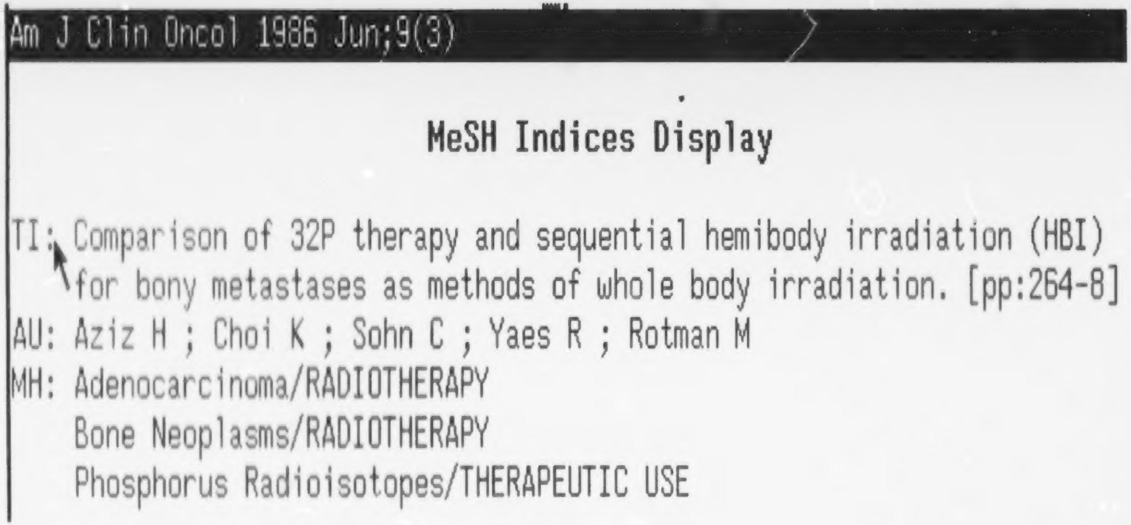


Am J Clin Oncol 1986 Jun;9(3)

MeSH Indices Display

TI: Comparison of 32P therapy and sequential hemibody irradiation (HBI) for bony metastases as methods of whole body irradiation. [pp:264-8]
AU: Aziz H ; Choi K ; Sohn C ; Yaes R ; Rotman M
MH: Adenocarcinoma/RADIOTHERAPY
Bone Neoplasms/RADIOTHERAPY

Figure 13. Sample MeSH indices display window (MedIndEx) (1)



```
Am J Clin Oncol 1986 Jun;9(3)

                    MeSH Indices Display

TI: Comparison of 32P therapy and sequential hemibody irradiation (HBI)
    for bony metastases as methods of whole body irradiation. [pp:264-8]
AU: Aziz H ; Choi K ; Sohn C ; Yaes R ; Rotman M
MH: Adenocarcinoma/RADIOTHERAPY
    Bone Neoplasms/RADIOTHERAPY
    Phosphorus Radioisotopes/THERAPEUTIC USE
```

Figure 14. Sample MeSH indices display window (MedIndEx) (2)

```
(|Medical Subjects|)
(|Disease|                      INHERITS-FROM
                                (VALUE |Medical Subjects|))
(|Pathological Structures|      INHERITS-FROM
                                (VALUE |Medical Subjects|))
(|Neoplasms|                    INHERITS-FROM
                                (VALUE |Disease| |Pathological Structures|))
(|Neoplasms by Site|            INHERITS-FROM
                                (VALUE |Neoplasms|))
(|Musculoskeletal Diseases|     INHERITS-FROM
                                (VALUE |Disease|))
(|Bone Diseases|                INHERITS-FROM
                                (VALUE |Musculoskeletal Diseases|))
(|Bone Neoplasms|               INHERITS-FROM
                                (VALUE |Bone Diseases| |Neoplasms by Site|))
(|Femoral Neoplasms|            INHERITS-FROM
                                (VALUE |Bone Neoplasms|))
```

Figure 15. Femoral Neoplasms INHERITS-FROM ancestry (MedKB)

Femoral Neoplasms (with the top-level frame *Medical Subjects*, from which all frames ultimately inherit). In reality, the *Bone Neoplasms* MedKB frame (Figure 16) has no domain-specific slots, but only INHERITS-FROM and CHILDREN, in contrast to the earlier version of this frame (Figure 4). We now see that assistance the system provides in filling in *Bone Neoplasms* indexing frames is not encoded in *Bone Neoplasms* after all, but instead higher up in the inherits-from hierarchy, specifically slot facets in *Bone Diseases* or *Neoplasms by Site*—the INHERITS-FROM value here—or perhaps a still higher frame, such as *Neoplasms* or *Disease*.

MEDKB MANAGER

Managing MedKB, that is, creating and editing knowledge-base frames, is more complicated than managing a thesaurus, since the knowledge engineer (formerly known as the thesaurus specialist) is responsible not only for the terms, but also for encoding data and procedures needed for providing interactive indexing assistance, and for keeping track of and using inheritance. The system, in effect, merges a thesaurus and indexing manual, in a potentially concise, executable form.

Modifying MedKB is essentially the process of editing facets. Two requirements of a knowledge base are that it be consistent and have proper syntax. We have developed an interactive Knowledge Base Management System (KBMS), known as MedKB Manager, designed to meet these requirements. General functions performed using MedKB Manager are summarized as follows:

- Creating new frames (adding frame terms to CHILDREN slot, VALUE facet, of existing frames);
- Making inheritance links (adding frame terms to INHERITS-FROM slot, VALUE facet, of lower level frames);
- Encoding indexing assistance (modifying fillers of facets, such as RESTRICTIONS, IF-NEEDED, CAN-CONTINUE?, and so forth).

MedKB Manager ensures consistency by preventing a frame from being isolated from the rest of the knowledge base, and by using inheritance from ancestral frames for displaying slot- and facet-names and default fillers; it suggests fillers based on other fillers, and it can constrain fillers. The system ensures proper syntax as much as possible by using devices such as menus, template code, and cut and paste. In any event, frame syntax is ensured down through the facet name for all facets.

Syntax of contents of RESTRICTIONS facets is ensured by having the system display code in the form of menus, and conversely by having the system generate code based on users' menu selections. Creating and editing contents of IF-NEEDED facets are facilitated by pop-up menus, user selection of template code (possibly requiring minor modification), and cut and paste. Other facets are modified using an editor into which

```
(|Bone Neoplasms|
  (INHERITS-FROM
     (VALUE  |Bone Diseases| |Neoplasms by Site|)
     (I-PARENT
        (BODY-SITE  (RESTRICTIONS |Bone Diseases|)
                    (SPECIFICITY |Bone Diseases|)
                    (DEFAULT |Bone Diseases|))
        (BIOLOGICAL-FINDING
                    (RESTRICTIONS |Neoplasms by Site|)))))
  (CHILDREN
     (VALUE  (|Medical Subjects| |Femoral Neoplasms|))
     (SPECIFICITY (BODY-SITE (|Femoral Neoplasms|))))))
```

Figure 16. Sample Bone Neoplasms MedKB frame

the system writes files that contain default fillers and helpful comments. This being a LISP editor, it evaluates code in these files, thereby catching syntax errors that might otherwise cause the system to bomb. Ultimately, we hope to extend to all facets—not just RESTRICTIONS and IF-NEEDED—techniques that will tend to free knowledge engineers from being expert programmers.

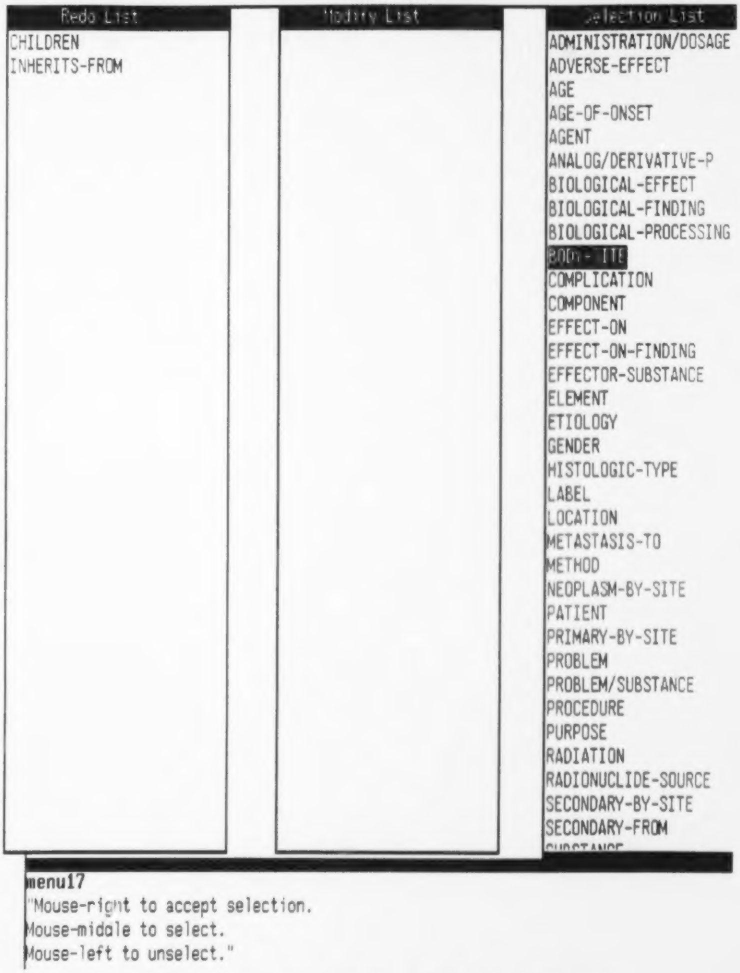
To illustrate, a new frame, *Vitamins*, might be created by first adding this term to the CHILDREN slot, VALUE facet, of two existing frames, *Biological Substances* and *Drugs*. Editing the new *Vitamins* frame itself would begin by filling the VALUE facet in its INHERITS-FROM slot; MedKB would display as default fillers the terms corresponding to frames in which *Vitamins* was added as a child.

Updated INHERITS-FROM and CHILDREN hierarchies can be viewed as soon as modifed frames resulting in these new versions of hierarchies have been written to temporary files during a MedKB Manager session. Therefore, by modifying very high-level frames, one can make rather radical changes in one fell swoop, and then change them back if they are not right. This procedure is much less cumbersome than assigning numerical codes (e.g., MeSH tree numbers) for hierarchies.

To illustrate use of menus to generate code, we will create the RESTRICTIONS facet for the BODY-SITE slot in the top-level frame *Disease* which inherits from *Medical Subjects*. Since *Medical Subjects* does not contain domain-specific slots, top-level frames, like *Disease*, must have these slots created in them explicitly.

After the *Disease* frame has been specified for editing, and the user has indicated that a slot is to be modified, the next step is selecting the slot to be modified. Figure 17 shows menu lists of all available slots. The Redo List window contains immediate slots that have already been created: CHILDREN and INHERITS-FROM. The Modify List window at this point normally contains inherited slots, which may be selected for modification, in which case they will override inheritance and be transferred to Redo List. Remaining slots, in the Selection List window, may be selected and transferred to Modify List. In this figure, BODY-SITE has been selected for editing.

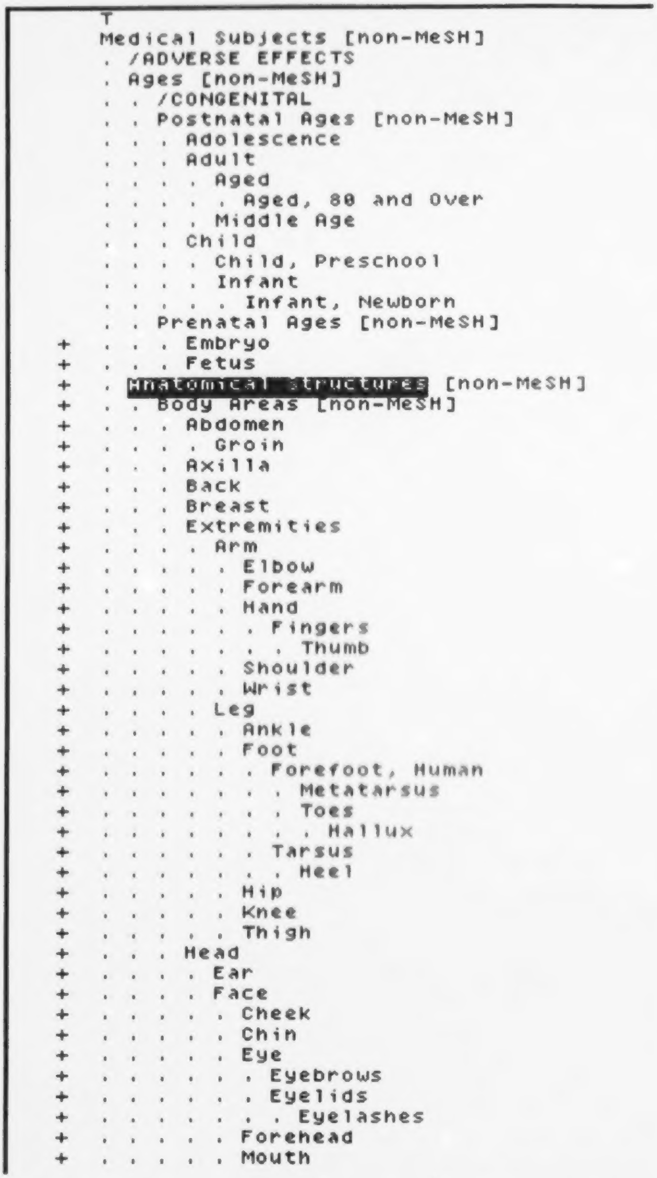
Once a slot has been selected, the next step is to select the facet to be edited. After selecting the RESTRICTIONS facet, a system message will remind the user that no inherited filler is available, which means that permissible fillers (for contents of this facet) must be selected from a display of the entire MedKB hierarchy. Figure 18 shows the first page of the MedKB hierarchical display, where the user has moused on *Anatomical Structures* to select this hierarchy as permissible fillers for body site of disease; the plus sign marks terms and hierarchies the user has selected.


Redo List
CHILDREN
INHERITS-FROM
Modify List
Selection List
ADMINISTRATION/DOSAGE
ADVERSE-EFFECT
AGE
AGE-OF-ONSET
AGENT
ANALOG/DERIVATIVE-P
BIOLOGICAL-EFFECT
BIOLOGICAL-FINDING
BIOLOGICAL-PROCESSING
BODY-SITE
COMPLICATION
COMPONENT
EFFECT-ON
EFFECT-ON-FINDING
EFFECTOR-SUBSTANCE
ELEMENT
ETIOLOGY
GENDER
HISTOLOGIC-TYPE
LABEL
LOCATION
METASTASIS-TO
METHOD
NEOPLASM-BY-SITE
PATIENT
PRIMARY-BY-SITE
PROBLEM
PROBLEM/SUBSTANCE
PROCEDURE
PURPOSE
RADIATION
RADIONUCLIDE-SOURCE
SECONDARY-BY-SITE
SECONDARY-FROM
menu17

```
"Mouse-right to accept selection.
Mouse-middle to select.
Mouse-left to unselect."
```

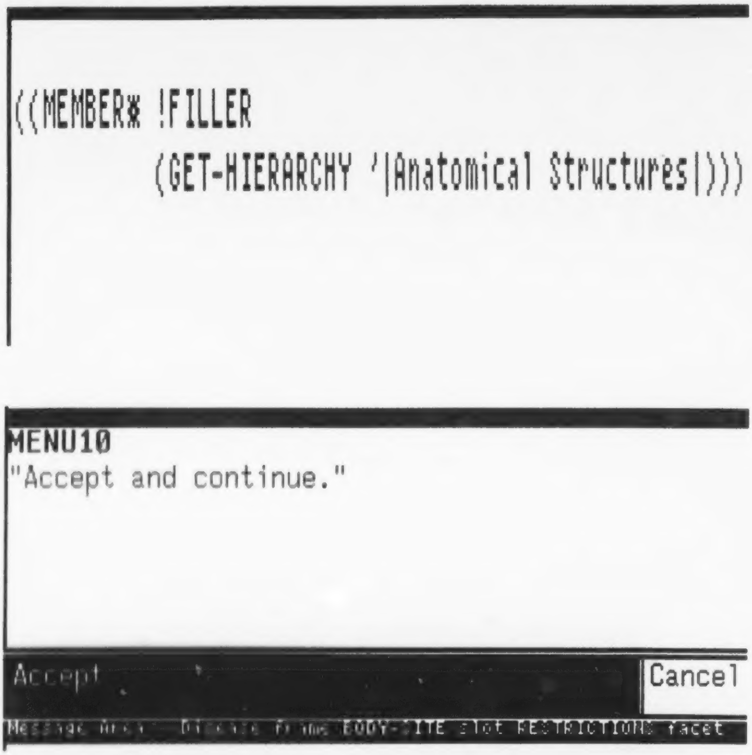
Figure 17. Menu lists of all available slots (MedKB)

The system then analyzes selections and actually writes code corresponding to them, as shown in the top window in Figure 19. When the system finishes encoding, it permits the user to edit the facet contents, which have been written to a file and placed in a screen editor, even though this is unnecessary (and in fact not recommended, except to



```
       T
       Medical Subjects [non-MeSH]
       . /ADVERSE EFFECTS
       . Ages [non-MeSH]
       . . /CONGENITAL
       . . Postnatal Ages [non-MeSH]
       . . . Adolescence
       . . . Adult
       . . . . Aged
       . . . . . Aged, 80 and Over
       . . . . Middle Age
       . . . Child
       . . . . Child, Preschool
       . . . . Infant
       . . . . . Infant, Newborn
       . . Prenatal Ages [non-MeSH]
   +   . . . Embryo
   +   . . . Fetus
   +   . Anatomical Structures [non-MeSH]
   +   . . Body Areas [non-MeSH]
   +   . . . Abdomen
   +   . . . . Groin
   +   . . . Axilla
   +   . . . Back
   +   . . . Breast
   +   . . . Extremities
   +   . . . . Arm
   +   . . . . . Elbow
   +   . . . . . Forearm
   +   . . . . . Hand
   +   . . . . . . Fingers
   +   . . . . . . . Thumb
   +   . . . . . Shoulder
   +   . . . . . Wrist
   +   . . . . Leg
   +   . . . . . Ankle
   +   . . . . . Foot
   +   . . . . . . Forefoot, Human
   +   . . . . . . . Metatarsus
   +   . . . . . . . Toes
   +   . . . . . . . . Hallux
   +   . . . . . . Tarsus
   +   . . . . . . . Heel
   +   . . . . . Hip
   +   . . . . . Knee
   +   . . . . . Thigh
   +   . . . Head
   +   . . . . Ear
   +   . . . . Face
   +   . . . . . Cheek
   +   . . . . . Chin
   +   . . . . . Eye
   +   . . . . . . Eyebrows
   +   . . . . . . Eyelids
   +   . . . . . . . Eyelashes
   +   . . . . . Forehead
   +   . . . . . Mouth
```

Figure 18. First page of MedKB hierarchical restrictions display



```
((MEMBER* !FILLER
        (GET-HIERARCHY '|Anatomical Structures|)))
```

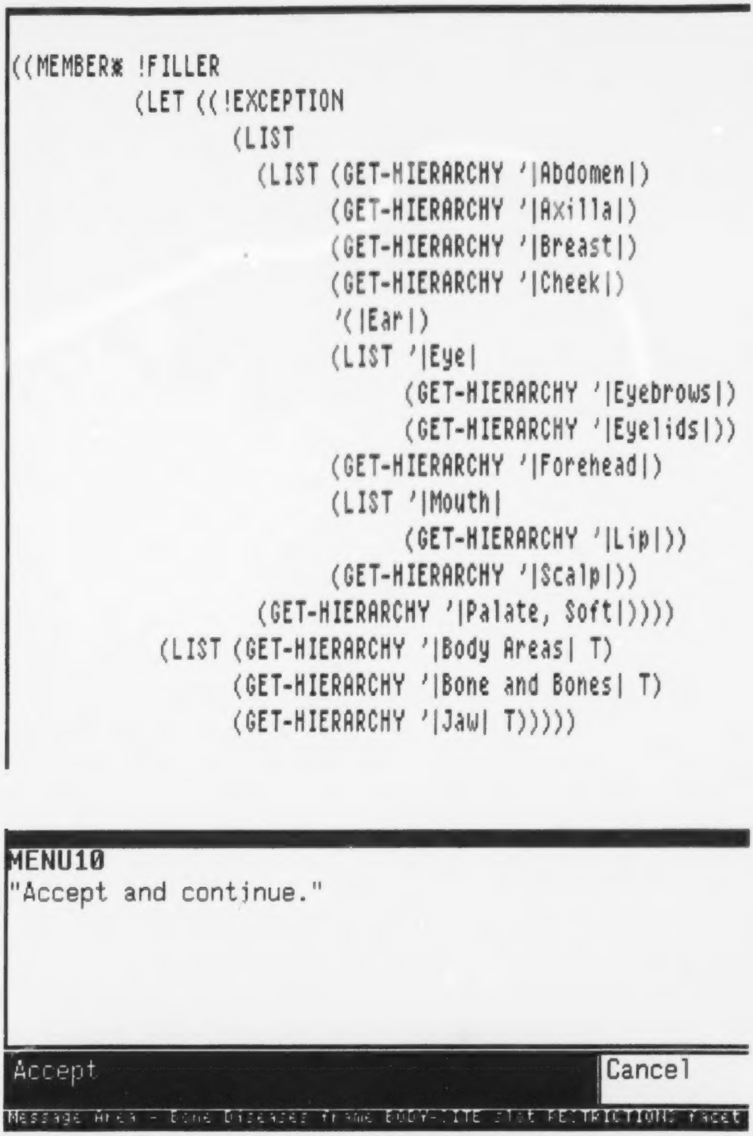
```
MENU10
"Accept and continue."
```

Accept
Cancel
Message Area - Disease frame BODY-SITE slot RESTRICTIONS facet

Figure 19. System-generated code for BODY-SITE of disease (MedKB)

overcome a system defect). This particular system-generated code is fairly simple, consisting of a membership function that checks if a filler being added to the BODY-SITE slot in indexing frames inheriting this slot is in the *Anatomical Structures* hierarchy.

Now that restrictions on body site have been encoded for the *Disease* frame, when the user elects to modify this same facet in the *Bone Diseases* frame, since *Bone Diseases* inherits from *Disease,* the system automatically displays a menu corresponding to inherited restrictions, which is just the *Anatomical Structures* hierarchy. From this menu, the user ultimately selects only hierarchies and terms making sense for bone diseases, namely, the bone hierarchy and certain body area terms elsewhere in the display. The top window in Figure 20 shows system-generated code based on these selections. This code is somewhat more



```
((MEMBER* !FILLER
        (LET ((!EXCEPTION
               (LIST
                 (LIST (GET-HIERARCHY '|Abdomen|)
                       (GET-HIERARCHY '|Axilla|)
                       (GET-HIERARCHY '|Breast|)
                       (GET-HIERARCHY '|Cheek|)
                       '(|Ear|)
                       (LIST '|Eye|
                            (GET-HIERARCHY '|Eyebrows|)
                            (GET-HIERARCHY '|Eyelids|))
                       (GET-HIERARCHY '|Forehead|)
                       (LIST '|Mouth|
                            (GET-HIERARCHY '|Lip|))
                       (GET-HIERARCHY '|Scalp|))
                 (GET-HIERARCHY '|Palate, Soft|))))
          (LIST (GET-HIERARCHY '|Body Areas| T)
                (GET-HIERARCHY '|Bone and Bones| T)
                (GET-HIERARCHY '|Jaw| T)))))
```

```
MENU10
"Accept and continue."
```

Accept
Cancel
Message Area - Bone Diseases frame BODY-SITE slot RESTRICTIONS facet

Figure 20. System-generated code for BODY-SITE of Bone Diseases (MedKB)

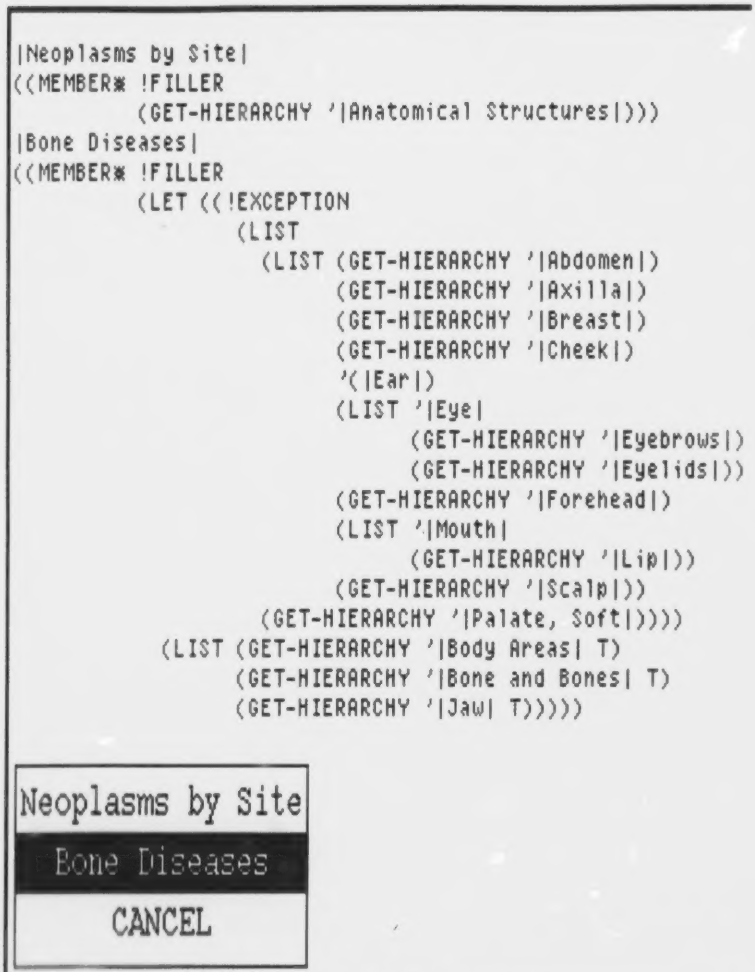
complicated than just a simple hierarchy as was the case for body site of *Disease*.

For the final example in this series, we focus on body site restrictions for *Bone Neoplasms,* which inherits from two frames—*Bone Diseases* and *Neoplasms by Site*. When the user indicates readiness to modify the RESTRICTIONS facet for this frame, the system displays code inherited from both inheritance parents, as well as a pop-up menu containing the names of these parents (Figure 21). If nothing is to be changed—that is, if inheritance should prevail—the default during indexing would be to inherit the union of restrictions from both parents. To relate this to the situation during an indexing session where the indexer is indexing a document about bone neoplasms, this would allow any anatomical structure permissible for *Neoplasms by Site* (for instance, *Heart* or *Liver*) to be entered as a body site structure in *Bone Neoplasms* indexing frames. Clearly this is undesirable, and we want, instead, for body site restrictions just from *Bone Diseases* to apply. Therefore, the user selects *Bone Diseases* as the frame from which body site restrictions should be inherited.

Figure 22 shows the pertinent portion of the final modified frame, which is *(BODY-SITE (RESTRICTIONS |Bone Diseases|)),* in the INHERITS-FROM slot, I-PARENT facet. Instead of explicitly repeating the same code in BODY-SITE of *Bone Neoplasms,* this expression points to restrictions in the selected inherits-from parent *Bone Diseases* which then causes the indexing system to access code in that location when checking or displaying body-site restrictions for *Bone Neoplasms* indexing frames. Thus, we see again that inheritance is used for avoiding redundancy.

In addition to ensuring proper syntax and economy, a major advantage of this menu interface is that it is not possible to make the mistake of inserting a noninheritable term into a restrictions list. Furthermore, system-generated code will always result in proper restrictions displays for the Indexer Interface, which may be quite difficult to ensure based on direct, human editing of MedKB.

As a shortcut for creating new hierarchies, MedKB Manager has a module for batch processing. Figure 23 shows the input file to create frames for the *Cardiovascular Diseases* hierarchy in batch mode, where MeSH tree numbers are used as hierarchy codes. After the batch job is complete, MedKB will contain bare frames, with only INHERITS-FROM and CHILDREN links, for new terms *Cardiovascular Diseases, Heart Diseases, Heart Neoplasms,* and *Vascular Diseases*. The remaining terms (*Disease, Neoplasms,* and *Neoplasms by Site*) are presumed to be in MedKB already, and the system will automatically add new VALUE fillers to their CHILDREN slot, as appropriate: in this case *Cardiovascular Diseases* is a new child of *Disease*, and *Heart Neoplasms*



```
|Neoplasms by Site|
((MEMBER* !FILLER
          (GET-HIERARCHY '|Anatomical Structures|)))
|Bone Diseases|
((MEMBER* !FILLER
          (LET ((!EXCEPTION
                  (LIST
                    (LIST (GET-HIERARCHY '|Abdomen|)
                          (GET-HIERARCHY '|Axilla|)
                          (GET-HIERARCHY '|Breast|)
                          (GET-HIERARCHY '|Cheek|)
                          '(|Ear|)
                          (LIST '|Eye|
                                (GET-HIERARCHY '|Eyebrows|)
                                (GET-HIERARCHY '|Eyelids|))
                          (GET-HIERARCHY '|Forehead|)
                          (LIST '|Mouth|
                                (GET-HIERARCHY '|Lip|))
                          (GET-HIERARCHY '|Scalp|))
                    (GET-HIERARCHY '|Palate, Soft|))))
            (LIST (GET-HIERARCHY '|Body Areas| T)
                  (GET-HIERARCHY '|Bone and Bones| T)
                  (GET-HIERARCHY '|Jaw| T)))))
```

Figure 21. Pop-up menu containing names of both inheritance parents (MedKB)

is a new child of *Neoplasms by Site.* At completion, the user can elect to be in interactive mode to further edit the frames in this hierarchy as necessary. The *Construction* hierarchy was taken from the *Thesaurus of ERIC Descriptors* to illustrate an alternative batch input format using dots rather than hierarchy code.

```
(|Bone Neoplasms| (PARENT (VALUE |Bone Diseases|
                                 |Neoplasms by Site|))
                  (INHERITS-FROM (VALUE |Bone Diseases| |Neoplasms by Site|)
                                 (I-PARENT
                                   (BODY-SITE (RESTRICTIONS |Bone Diseases|)
                                              (DEFAULT |Bone Diseases|)
                                              (SPECIFICITY |Bone Diseases|))
```

Figure 22. I-PARENT facet in sample Bone Neoplasms frame (MedKB)

In conclusion, development of a KBMS is a necessary offshoot of the MedIndEx System as it greatly facilitates the ordinarily complex task of creating and editing a rich knowledge base to maintain consistency and proper syntax—a task that is essential for research on the indexing part of the system. MedKB Manager can also be viewed as a knowledge acquisition tool to assist domain experts in building a knowledge base, particularly if interfaces are developed that do not require them to be expert programmers. In addition, heuristics may be developed to further automate knowledge-base creation.

A Note on Retrieval

As stated earlier, an output of MedIndEx is conventional MeSH indexing terms, which might then become part of the regular MEDLINE database for retrieval.

At the moment there is no standard retrieval language for searching frame databases, just as there is no standard frame language. One might write such a retrieval language, and develop an expert retrieval system using MedKB as its knowledge base. Or one might develop an expert system using MedKB that would index search queries, in contrast to documents, thereby suggesting conventional MeSH search terms. The difficulty would be devising or implementing strategies once a set of search terms has been produced.

Another approach, which we have begun to investigate, is automatically translating frame databases into relational databases, the advantage being the use of commercially available relational database management systems (RDBMS) including Search Query

From MeSH Tree Structures

```
000 Disease
000.C14 Cardiovascular Diseases
000.C14.280 Heart Diseases
000.C14.280.459 Heart Neoplasms
000.C14.907 Vascular Diseases
000.C04 Neoplasms
000.C04.588 Neoplasms by Site
000.C04.588.894 Thoracic Neoplasms
000.C04.588.894.309 Heart Neoplasms
```

From ERIC Two-Way Hierarchy Term Display

```
Construction (Process)
. . Cabinetmaking
. . Carpentry
. . Masonry
. . . . Bricklaying
. . Prefabrication
. . Road Construction
. . School Construction
```

Figure 23. Batch input for creating MedKB hierarchies

Language (SQL). However, it would then be necessary to encode the knowledge-base hierarchy in the RDBMS and write a front-end to SQL to facilitate retrieval.

HARDWARE, OPERATING SYSTEM, COMPUTER LANGUAGE, AND SYSTEM SIZE

It seems obvious that an expert system such as MedIndEx would be impossible if not for the workstation environment. This is provided by the Sun SPARCstation™ under the SunOS ® operating system using the SunView™ window environment.

MedKB and other software unrelated to running the system interface are written in an experimental frame language, Framer, developed on top of Sun Common Lisp 3.0™ which, without using special extensions, is virtually identical to standard Common Lisp. Part of MedKB Manager is written in CLOS (Common Lisp Object System), which also runs on various hardware and operating systems. The interface is written in the Window Tool Kit of Sun Common Lisp, which uses SunView and therefore is machine dependent. Machine and operating system dependency would be eliminated if the interface were rewritten in the standard X Windows.

The system contains 1,400 frames and thirty-seven slots, with a range of one to eight fillable slots for a frame. The size of Common Lisp and CLOS executable code is 14.4 megabytes. The system itself is 4 megabytes, 1 megabyte for MedKB and related files (Word File and Alias Table); if 5,000 frames (about one-third of MeSH descriptors), the system might be about 7.6 megabytes; 15,000 frames, perhaps 14.7.

Commercial expert system environments and microcomputer-based shells were not available to us at the start of the MedIndEx project in late 1986. Use of three different shells on the IBM PC/XT was reported by Sharif (1988) for developing an expert system for subject classification of monographs, specifically, selecting Dewey Classification numbers from a small section of the DC19 schedules. Her report concludes that shells, although suitable for many applications, are not appropriate for developing large-scale expert systems in classification, and recommends developing an expert system *ab initio* using languages such as Prolog or LISP in preference to expert system tools. Reasons for unsuitability include limited size capacity and insufficient flexibility of knowledge representation structures. (Illustrating presumably more appropriate use of shells for rapid prototyping, within two days an advisory system was developed using one of the AACR2 algorithms for choice and form of access point. Incorporating additional rules covering form of heading for corporate body increased the knowledge base to 90 kilobytes. Although response time on an IBM PC/XT was acceptable, loading of the knowledge base took two to three minutes. It should be noted that the system did not provide a catalog entry, but rather guided the user to relevant sections of AACR2.)

CONCLUSIONS

Factors contributing to support of AI techniques include existing thesaurus-based indexing systems, in particular if they are interactive, and organizational commitment to enhancing and expanding this approach.

These apply to NLM, evidenced by MeSH as the basis for retrieval for most NLM databases, of which MEDLINE, covering nearly twenty-five years of literature, is the most well known. MeSH is an exceptionally well-controlled thesaurus, with full-time staff completely dedicated to its management. It is updated annually using a thesaurus management system, MeSH 204 (running on the Model 204 ® database management system), to ensure consistency and proper syntax. Documentation of indexing rules and policy by the *MEDLARS Indexing Manual* and *Technical Notes*, as well as MeSH annotations, is a mainstay of the indexing system. The online Automated Indexing and Management System (AIMS) validates indexers' entries, checking for misspelled terms and invalid main heading-subheading precoordinations, and may add terms automatically or display system warnings consequent to certain previous entries by the indexer. Plans are underway to improve this interface, which serves as a good transition to a knowledge-based indexing system.

Looking toward the future, an important program at NLM is the Unified Medical Language System (UMLS™) project, which will include Metathesaurus™, a machine-readable knowledge source containing information about biomedical concepts and their representation in different vocabularies and thesauri; it will represent a variety of relationships among terms and support mapping from users' terms to appropriate controlled vocabularies and among different controlled vocabularies.

NLM's commitment to the use of controlled vocabularies and knowledge bases extends to the projects of its research divisions, the Lister Hill National Center for Biomedical Communications (LHNCBC) and recently created National Center for Biotechnology Information (NCBI). LHNCBC research projects include the Natural Language Systems Project, which is developing SPECIALIST, an experimental system for parsing, analyzing, and accessing biomedical text; the parsing system requires an extensive and well-specified lexicon with explicit links to a knowledge base of biomedical concepts. In building and enhancing databases of genomic information, a goal of NCBI is a common search vocabulary for retrieving genetic sequence records from GenBank, the national DNA sequence database, and retrieving MEDLINE literature referencing the same concepts.

MedIndEx, an LHNCBC project, is part of this research agenda based on a long-standing, continuing commitment to thesaurus- and knowledge-based approaches to facilitate and enhance user access to biomedical information.

ACKNOWLEDGMENT

Thanks to De-Chih Chien of Management Systems Designers, Vienna, Virginia, for his work in designing, writing, and installing the interface for the MedIndEx System.

REFERENCES

Humphrey, S. M. (1989a). Knowledge-based expert system for computer-assisted indexing. *IEEE Expert, 4*(3), 25-38.

Humphrey, S. M. (1989b). MedIndEx system: Medical indexing expert system. *Information Processing & Management, 25*(1), 73-88.

Humphrey, S. M., & Chien, D. (1990). *The MedIndEx system: Research on interactive knowledge-based indexing and knowledge management.* Bethesda, MD: National Library of Medicine. (NLM-LHC-90-03, NTIS No. PB90-234964/AS)

Humphrey, S. M., & Miller, N. E. (1987). Knowledge-based indexing of the medical literature: The Indexing Aid Project. *Journal of the American Society for Information Science, 38*(3), 184-196.

Sharif, C. A. Y. (1988). *Developing an expert system for classification of books using micro-based expert system shells.* British Library Research Paper 32. London: British Library Research and Development Department. Distributed by British Library Publications Sales Unit, Boston Spa, Wetherby, West Yorkshire LS23 7BQ, England.

JAMES R. PARROTT

Davis Centre Library
University of Waterloo
Waterloo, Ontario, Canada

Reference Expert Systems: Foundations in Reference Theory

1. INTRODUCTION

A reference expert system may be considered to be a system with a knowledge base covering various aspects of the reference process in a library setting. Knowledge bases generally consist of several components (such as databases, rule bases, frames, and semantic nets) that interact with an inference engine, a user interface, and each other. This paper will examine progressively more complex knowledge-based systems for reference that can be constructed from components like these, concentrating at first on combinations of databases and rule bases. This examination will lead to a classification of reference expert systems.

In Section 3, very simple architectures of a type common in other fields will first be considered. Arguments drawn from reference theory suggest that these simple architectures are appropriate primarily in dealing with directional reference transactions. In Sections 4 and 5, reference theory will be used to develop two additional architectures more appropriate to other reference transactions, such as ready-reference transactions. The classification of reference expert systems will be completed in Section 6 by examining further reference theory and then using it to develop variants on the three basic types. Section 7.1 will discuss briefly the use of reference knowledge bases for computer-assisted instruction. Section 7.2 will consider deep reference knowledge. The paper will conclude with some prognostications about future developments.

Fundamental to this paper is the premise that an examination of the design of reference expert systems may profitably be guided by the experience embodied in existing models of the reference process. This point of view, if not the exact analytical approach adopted here, has been expressed previously (Parrott, 1990). The classification that is based on these models was modified after presentation at the Clinic as a result of the suggestion there by Charles Bailey (University of Houston) and Lloyd Davidson (Northwestern University) that the scheme be extended to include hypertext systems. That led to a consideration of several other issues, including combinatorial completeness of the scheme. The end result is a classification significantly richer than that presented before the Clinic audience.

2. PRELIMINARY CONSIDERATIONS

2.1 Answers in the Reference Process

Because reference work is often a multistage process, with intermediate results before the desired information is obtained, the concept of answer can be somewhat ambiguous. At least three major types of answers can be distinguished:

1. **Desired Information.** For example, for a ready-reference question, an address might be the desired information. This type of information might be called the "final answer."
2. **Bibliographic Information** about reference tools, books, and other materials (printed or electronic) that experience has shown could contain the desired information. An example is bibliographic information about *Encyclopedia of Associations*. This kind of bibliographic information is generally referred to in this paper as "titles of information sources" (although more than the titles is intended), or abbreviated simply as "information sources." This kind of information may be thought of as an intermediate answer of the form: "The information you want is probably found in *Encyclopedia of Associations*."
3. **Categories of Information Sources** that experience has shown could contain the desired information. For example, the experience of many reference librarians is that trade directories are a useful category of information source giving addresses of manufacturers. This kind of information is referred to as "type of information source" in this paper. This kind of information may also be thought of as an intermediate answer, one especially useful when instruction is important. It may be considered to be an intermediate answer of

the form: "The kind of information you want is generally found in trade directories."

2.2 Differences Between Databases and Rule Bases

Databases and rule bases embody different ways of organizing knowledge. Suppose a piece of knowledge about reference consists of certain descriptors (subject, geographical area, etc.) and associated answers (hours of opening, biographical information, etc.). A database approach puts that knowledge into records, with fields for the descriptors and fields for the answers. Below, several approaches to storing information in databases are distinguished.

1. A page-based approach, where the answer field contains some text and answers (possibly enough to fill a page or screen); the answers are not labeled as such to distinguish them from the text. In general, more than one answer is found in each answer field; conversely, one answer may appear in more than one record. For example, a record might give hours of opening for several branch libraries, or refer to several reference tools; the answers will not be labeled to distinguish them from other text (such as "The following are the hours of opening...").
2. A hypertext-based approach, where the answer field contains some text and answers (possibly enough to fill a page or screen); the answers are labeled to distinguish them from the text. The situation is identical to the page-based approach, except that individual answers are now labeled.
3. A single-answer-based approach, where the answer field contains some text and *one* answer. The answer is not labeled to distinguish it from the text. The situation is identical to the page-based approach, except that the answer field for a record contains only a single answer, which appears, moreover, only in that one record.
4. An item-based approach, where the answer field for a record contains nothing but one answer, which appears, moreover, only in that one record. For example, a record will give hours of opening for only one branch library, or refer to only one reference tool. A variant of this allows the answer field to have text, but requires the answer to be labeled.

A rule-based approach puts that knowledge into rules: the descriptors in the IF clauses and the answers in the THEN clauses. It is quite possible for the same answer to appear in the THEN clauses of several rules.

Notice two critical features in the above:

a. The possibility that *a single answer appears in several places* *in a file.* If this is so, then it may be considered a disadvantage, since updating the answers will not necessarily be easy. But it may also be considered an advantage, as the following argument shows. Suppose we attach weights (confidence factors) to each answer. That is, for the set of descriptors, we have X confidence that the answer will be useful. But if a single answer may appear in several places in the file, then we can assign it several different weights: one for each set of question descriptors. This is a much more realistic way of assigning weights to answers than simply assigning one weight to each. So, if a single answer may appear in several places (as in a rule base, a hypertext database, or a page-based database), updating may not be easy, but realistic weights are possible. Conversely, if a single answer appears in only one place (as in a single-answer-based database or an item-based database), then updating is easy, but realistic weights are not possible.

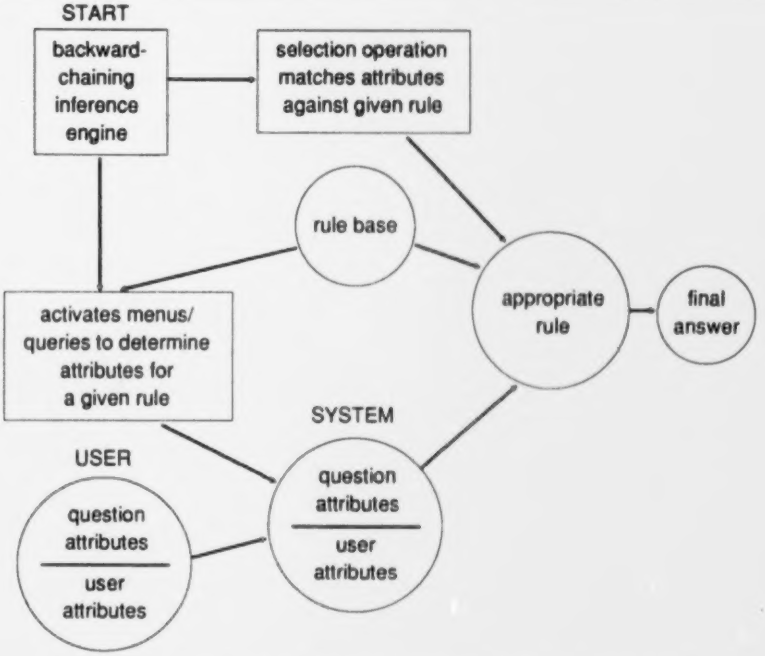

START
backward-chaining inference engine
selection operation matches attributes against given rule
rule base
appropriate rule
final answer
activates menus/ queries to determine attributes for a given rule
SYSTEM
USER
question attributes
user attributes
question attributes
user attributes

Figure 1. A type 1R model with a backward-chaining rule base

There is a further point that needs to be considered. If, indeed, a single answer appears in several places in a file, rather than one, then the file may serve to eliminate possibilities in the following sense. Suppose, first, that a single answer can appear in only one place in the file. Then that answer will have a fixed set of descriptors attached to it. But if it could occur in several places in the file, it could have different sets of descriptors associated with it. Now, assume that these sets of descriptors are consistent with that single set in the former arrangement. But these sets do not need to exhaust all the possibilities of the former arrangement.

For example, suppose in the former arrangement that a tool is assigned descriptors such as: *question-type* = biographical, *geographical-area* = U.S.A., *subject-area* = chemistry, *sector* = academic, *alive-or-dead* = alive. But in the latter arrangement, there might be one place where the given answer is assigned only *question-type* = biographical, *geographical-area* = U.S.A., and *subject-area* = chemistry. And the only other occurrence of the answer might have *question-type* = biographical, *sector* = academic, and *alive-or-dead* = alive. Consider a question with question attributes: *question-type* = biographical, *subject-area* = chemistry, *alive-or-dead* = alive. These question attributes will match the correct tool in the first arrangement, but not in the second, since neither of the two occurrences of that tool in the second arrangement have the given cluster of attribute/values.

b. The possibility of identifying the answer inside the answer field with precision. If this is so (as in an item-based database, a hypertext database, or an appropriately constructed rule base), then (1) it is possible to link a particular answer up with an external database, which will be considered later on, and (2) it is possible to assign a weight directly to a particular answer in the answer field, even if there are several answers in that field (that is, we have more realistic weighting than otherwise). Unlike the feature previously considered, there is an advantage only when this feature has a positive value (when precision exists). When this feature has a negative value (when precision is lacking), then there is no advantage.

We have thus identified two important features, each with two values. Of the resulting four values, three are advantageous in certain situations.

3. SYSTEMS WITH ONE SELECTION OPERATION

This section treats the simplest structures possible for a knowledge-based system. These structures may have components like a database, a rule base, or a combination of both. But in these simple structures, the final answer is determined by a match (or selection operation) of the question attributes against *only one* of these components. Systems using only one selection operation in this manner will be called Type 1 systems.

Although, as will be seen, these structures have significant limitations, there are several motivations for beginning with structures as simple as this. The first motivation is a pedagogical one: applications of databases and rule bases as simple as this are easy to understand. The second is a practical one: basic structures like this are easy to implement using available software. The third motivation is an imitative one: these structures have proven useful in other fields.

3.1 Type 1 Systems with One Component and Realistic Weights

As noted in Section 2.2, realistic weights are possible in components such as a rule base (R), a hypertext database (H), and a page-based database (P). Systems built from only one of these components may be called 1R, $1H_s$, and $1P_s$ systems, respectively. (The subscript s stands for matching against *scope* attributes.) Updating will not necessarily be easy in systems like this.

Suppose one has a Type 1R system. As an example of its operation, consider a transaction in which a user wants to know the hours of opening of a particular branch library, the Botany Library. Suppose further that the rule base contains the following rule:

> IF the question type = hours-of-opening and the branch-library = botany THEN the answer is "Monday to Friday, 8:30 AM to 10:00 PM; Saturday and Sunday, 1:00 PM to 6:00 PM."

If the inference engine uses backward chaining, then it will pick its rules one by one, and ask the user questions to determine the values of the attributes. When the engine reaches the rule above, if the user has not already revealed the value of *question-type* and *branch-library*, the inference engine will ask the user for these values. If the values match *hours-of-opening* and *botany* respectively, then the answer given in the THEN clause will be quoted. If the values do not match, another rule will be examined and more questions asked, if necessary.

If, on the other hand, the inference engine uses forward chaining, then it will ask the user a series of questions (using either a set of menus or frames), and then do a match against the entire rule base. If the user has given *hours-of-opening* as the *question-type* and *botany* as the *branch-library*, then a match is obtained on the rule mentioned above. The answer given in the THEN clause will be quoted. No more questions need to be asked, since they have all been asked at the beginning.

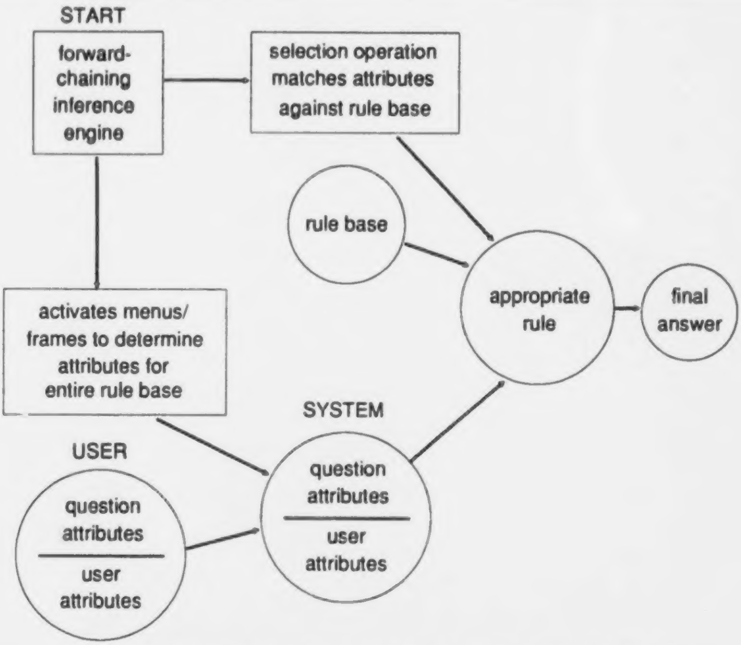

START
forward-chaining inference engine
selection operation matches attributes against rule base
rule base
appropriate rule
final answer
activates menus/ frames to determine attributes for entire rule base
SYSTEM
question attributes
user attributes
USER
question attributes
user attributes

Figure 2. A type 1R model with a forward-chaining rule base

In Type $1H_s$ and $1P_s$ systems, the internal operation will differ from that of Type 1R systems. But all three cases share the following features:

1. the question attributes are somehow determined from the user;
2. they are matched against either the IF clauses of rules in a rule base, or the descriptors in a page-based database or a hypertext database;
3. the answer is in the THEN clause of the rules, or the answer field in the database; and
4. a given answer may appear in more than one place in the rule base or database, hence allowing for realistic weights, but also allowing problems in updating the answers.

3.2 Type 1 Systems with One Component and Easy Updating

As noted in Section 2.2, easy updating of answers is possible in components such as a single-answer-based database (S) or an item-based database (I). Systems built from only one of these components may be called $1S_s$ and $1I_s$ systems, respectively. (Again, the subscript s stands for matching against scope attributes.) Realistic weights will not be possible in systems like this.

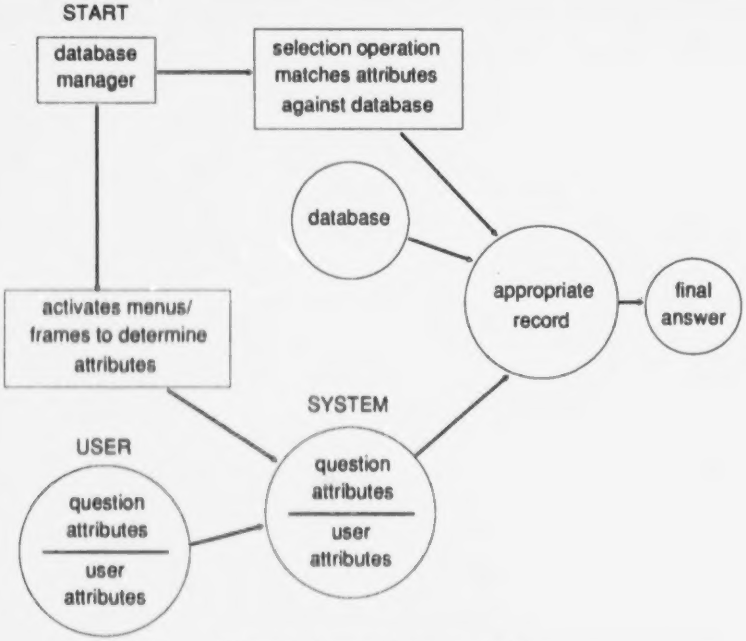

START
database manager
selection operation matches attributes against database
database
appropriate record
final answer
activates menus/ frames to determine attributes
SYSTEM
USER
question attributes
user attributes
question attributes
user attributes

Figure 3. A type $1I_s$ or $1S_s$ model

In systems like this, question attributes are matched against a database (S or I) of factual information. Since no matching is done against a rule base, the question attributes cannot be determined through backward chaining by an inference engine. As in forward chaining on a rule base, either a set of menus or frames must be used. Once the question attributes have been determined, their values are matched against the attributes (or the scopes) of the database records. The matched records are then used in constructing the display used to answer the question.

Suppose one has a $1I_s$ system. As an example of its operation, consider the same kind of question as before, namely, a transaction in which a user wants to know the hours of opening of a particular library branch, the Botany Library. Suppose further that the database contains a record including the following fields and values:

- question-type = hours-of-opening
- branch-library = botany
- answer = "Monday to Friday, 8:30 AM to 10:00 PM; Saturday and Sunday, 1:00 PM to 6:00 PM."

The question attributes gathered through the reference interview will match this record. The system will then display the contents of the answer field, which contains the hours of opening of the Botany Library.

The situation for a $1S_s$ system is the same, except that the answer field will contain not only the answer proper, but additional text as well, and the former is not labeled to distinguish it from the latter. For example: answer = "Hours of opening of the Botany Library: Monday to Friday, 8:30 AM to 10:00 PM; Saturday and Sunday, 1:00 PM to 6:00 PM."

3.3 Type 1 Systems with More Than One Component

In Section 2.2, we concluded that three values of features were advantageous in some situations: realistic weights, easy updating, and precision. Considered in Section 3.1 were Type 1 systems with realistic weights, but not easy updating; some had precision, some did not. In Section 3.2, we considered Type 1 systems with easy updating, but not realistic weights; again, some had precision, some did not. In both those sections, we looked at one-component systems. The question arises: Is it possible, by considering systems with more than one component, to generate the other combinatorial possibilities? In particular, can one construct Type 1 systems that have *both* realistic weights and easy updating or *neither*?

Examine the possibilities in two-component systems. The first case is a system constructed of two components each of which allows realistic weights but not easy updating. A bit of reflection shows that such a system is equivalent to the systems in Section 3.1. As an example, consider one in which the first component is a rule base and the second is a page-based database, denoted Type 1RP. The system will first determine the question attributes, perhaps using a set of menus. Then the attributes will be matched against the IF clauses of the rules in the rule base. The THEN clause of a matching rule will point to a page in the database. Since both the components allow the specification of realistic weights, the total system will certainly allow it too. But since neither component allows easy updating, the total system cannot allow it. Hence the system is equivalent to those in Section 3.1.

The second case is a system constructed of two components each of which allows easy updating, but not realistic weights. Similarly, such a system is equivalent to the systems in Section 3.2. The third case is a system whose first component allows easy updating but not realistic weights, and whose second component allows realistic weights but not easy updating. Such a system is equivalent to the systems in Section 3.1, since it clearly allows realistic weights but not easy updating (since the answer that must be updated lies in the second component).

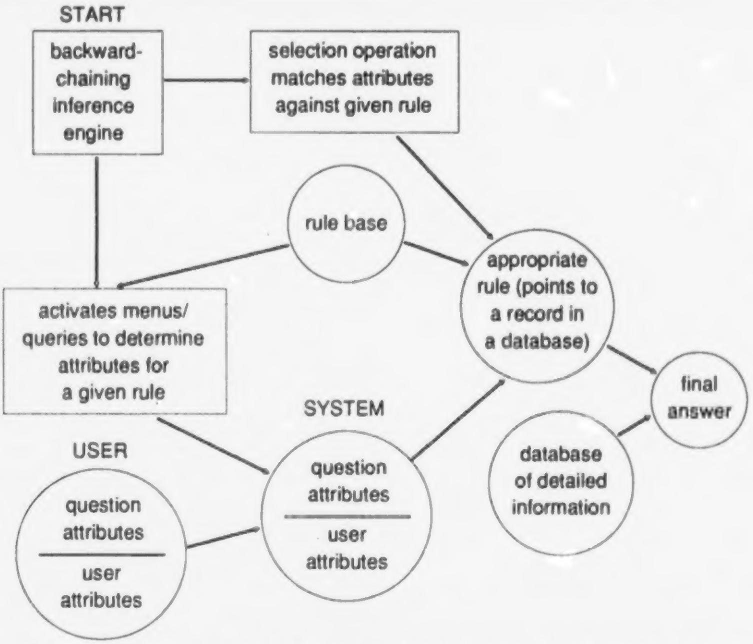

START
backward-chaining inference engine
selection operation matches attributes against given rule
rule base
appropriate rule (points to a record in a database)
activates menus/ queries to determine attributes for a given rule
final answer
SYSTEM
USER
question attributes
user attributes
question attributes
user attributes
database of detailed information

Figure 4. A type 1RI or 1RS model with a backward-chaining rule base

The fourth case is the interesting one. Here, the first component allows realistic weights but not easy updating, and the second component allows easy updating but not realistic weights. Such a system allows *both* realistic weights (since one of the two components allows it) *and* easy updating (since the answer to be updated lies in the second component, which allows easy updating). Before proceeding to examine this case in detail, it should be noted that we have not been able to generate the combination in which *neither* realistic weights and easy updating hold.

A system corresponding to the fourth case operates as follows. It matches question attributes against a rule base (or H database) as in Section 3.1, but does not store the final answers in the rules (or records of the H databases). Instead, each rule (or record) points to one or more entries in an I or S database of final answers, which is consulted in constructing the display used to answer the question. Note that the first component allows the assignment of realistic weights to the answers, and the second component allows easy updating of the answers.

Finally, note that P databases have been deliberately excluded as the first component in this type of system. Although a P database will allow realistic weights, it lacks precision in specifying the answer; hence it cannot make a proper connection with a second component. To see this, suppose that the system used a page-based database to determine pointers. But then any given page might have several pointers on it, with no clear indication (to the system) where on the page the pointers occur. It would then be impossible for the system to determine what in fact the pointers actually are; the connection to the second component would thus not exist.

Suppose we have a 1RI system. As an example of its operation, consider the same kind of question as before, namely, a transaction in which a user wants to know the hours of opening of a particular library branch, the Botany Library. Suppose further that the rule base (the first component) contains the rule:

> IF the question has certain attributes,
> THEN go to record Y in the database of items for the factual information.

Somehow, the rule base carries out the reference interview, determining the question attributes, which match the above rule. That rule points to a record in the database (the second component); the answer field of the record is then displayed, giving the hours of opening of the Botany Library.

All other types of systems like this (Types 1RS, 1HI, and 1HS), will also clearly allow both realistic weights and easy updating.

3.4 Comparison of Type 1 Models

In Section 2.2 were identified three advantageous features that a system might have: realistic weights, easy updating, and precision in specifying answers. Before comparing the various types of systems described in the last three sections, let us consider whether all three of these features are useful in a Type 1 system. These Type 1 systems correspond to a model of a particular type of reference transaction put forward by William Katz (1982, pp. 72-75), which may be called Case 1 of Automatic Retrieval. The basic idea here is that, in some transactions, after data gathering (the reference interview), the data are used to extract the final answer from the librarian's memory. No recourse to intermediate information sources or reference tools is necessary. Hence, transactions like this will generally be directional in nature, giving locations, hours of service, and so forth. But in directional questions, the answers are unlikely to involve uncertainty: if the librarian is unsure, she will check some source (and therefore go beyond her memory and the bounds of a Type 1 transaction). Yet if no uncertainty is involved, then there is little point in assigning weights (confidence factors). From this, it follows that weights are not particularly useful in Type 1 systems.

What about precision in specifying the answer? With precision, the answer field either labels the answers (to distinguish them from additional text) or contains only one answer with no additional text. Without precision, the answer field may contain additional text that, for example, might recapitulate the question attributes, or name the field (as in "The hours of opening of the Botany Library are . . ."). As noted in Section 3.3, precision is necessary in the first component of a two-component system; it is not necessary in the second component, which is where the final answer lies. It is difficult to imagine a situation when precision in the *final* answer is essential in a Type 1 system, since the information in the answer field will be processed by a human being (not another system component), and humans are easily able to parse the answer proper from additional text. In Type 2 and 3 systems, the information in this answer field is not necessarily going to be processed by a human, so this argument will not hold there. In Type 1 systems, however, precision in the final answer is irrelevant.

In conclusion, there is only one important feature distinguishing the performance of Type 1 systems: ease of updating. We may therefore compare our systems as follows:

1. Easy updating: Types $1I_s$, $1S_s$, 1RI, 1RS, 1HI, and 1HS.
2. Not easy updating: Types 1R, $1H_s$, and $1P_s$.

3.5 Implementations of Type 1 Models

The implementations identified below appear to be restricted to Type $1P_s$ and $1H_s$ models. Some of the systems categorized as Type $1P_s$ systems might, however, actually be of other kinds, such as Type $1I_s$. The situation is not entirely clear, since the system descriptions in the literature do not always provide adequate details of implementation.

1. An early system, REFLES, handles factual data such as data associated with directional transactions (Bivins & Palmer, 1980). It uses a page-based database indexed by subject, and hence is of Type $1P_s$. Bivins was associated with another system that handles factual information, REFLINK (Bivins & Eriksson, 1982), which uses a page-based database with access via a subject index or a tree structure of menus. It is also of Type $1P_s$.

2. The Reference and Information Station (Purdue University Undergraduate Library) has menu access to a page-based database of factual information for answering directional questions (Smith & Hutton, 1984; Smith, D., 1989). It is therefore of Type $1P_s$.
3. The Information Function (IF) at Carnegie-Mellon University provides (within the online catalog) menu access to page-based information on library announcements, locations, services, and tips in using the catalog (Diskin & Michalak, 1985). It is thus of Type $1P_s$.
4. ORA (Online Reference Assistance) at the University of Waterloo Library (Parrott, 1986), has menu and keyword access to page-based directional information and other features as well. It is thus of Type $1P_s$.
5. The Information Machine (Fadell & Myers, 1989) at the University of Houston Library has menu access to a database of pages. Its pages contain directional-type information (locations, times, regulations, phone numbers) and other features. So, it is of Type $1P_s$.
6. The Apple Library Tour (Ertel & Oros, 1989) uses a hypertext database to provide directional and other information. It is thus of Type $1H_s$.

4. SYSTEMS WITH TWO SELECTION OPERATIONS

Simple expert systems in many other fields are able to operate quite satisfactorily using Type 1 architectures, that is, they are able to do one match and then provide the final answer. In reference work, this type of direct provision of factual information (i.e., without recourse to a reference tool) will generally be confined to answering directional transactions—that is, requests for directions, information about local services, hours of opening, etc. Much of the expertise of a reference librarian, however, is in locating information sources that may contain the required information, rather than in knowing the required information itself. More complex architectures are needed for this; they may be combined with Type 1 architectures to allow directional questions, too.

The salient feature of these complex architectures, then, is that they allow information sources to be prescribed as intermediate answers before obtaining a final answer. From this, four parameters of system behavior of Type 2 systems emerge. To show this, we consider the original three advantageous features discussed in Section 2.2, and see which, if any, are valid in a Type 2 system (they all are). We then see whether any other feature might be advantageous (an additional feature is uncovered).

1. The fact that an information source may not contain the required

information introduces an element of uncertainty into these considerations that was absent in Section 3. *Realistic weights* may now be important. Indeed, the ability to rank information sources by likelihood of success is a mark of an experienced librarian.

2. *Ease of updating* may be a concern in Type 2 systems as well, since bibliographic data on an information source may change with time. As before, updating will be easiest when an intermediate answer appears in only one location.
3. If we want the system to use knowledge about particular intermediate answers (information sources) to perform actions, then the system must have *precise access* to that knowledge. That is, it must know that a certain string of characters in a field corresponds to the title of an information source. This will allow us, for example, to link up to an external online CD-ROM database. In the latter case, we will have a *full implementation;* if a person must leave the terminal to consult the tool, we will have a *partial implementation.*
4. In the previous three points, we have reconsidered the three advantageous features discussed in Section 2.2. Another advantageous feature, peculiar to information sources, may now be added. Information about information sources is generally more structured than a final answer. In particular, we may have *indexing attributes* which tell us which fields in the information source are indexed. This may be important, since it might affect search time. Also, if the system knows which fields in an information source are indexed, then it will be able to deduce how the source should be searched (e.g., "search index A on value b" or "browse for value b"). For a partial implementation, this will be given only as part of the prescription to the user; for a full implementation, it will allow the expert system some control over the second matching operation, that on the information source itself.

So, for some (if not all) complex architectures, important features of system behavior include: *realistic weights, ease of updating, precision,* and the *indexing attributes* of an information source. Sections 4 and 5 each consider a particular model (both derived from Katz) of more complex reference transactions in which information sources are to be consulted.

The first model to be considered may be called a Type 2 model. It is derived from Case 2 of *Katz's Automatic Retrieval Model* (Katz, 1982, pp. 72-75). The basic idea here is that, after information gathering (the reference interview), the data are used to determine one or more information sources that may contain the desired factual information; the sources are then consulted. This paradigm might apply to directional transactions where the librarian has to consult an information source (not necessarily cataloged; possibly an in-house publication). The paradigm also applies to ready-reference transactions and substantive transactions.

Before going any further, it is useful to note that, although the attributes of final answers in Type 1 systems are of one kind only, intermediate answers in Type 2 systems (information sources) may have two distinct kinds of attributes: *scope attributes* (subject, geographical-area, etc.) and *indexing attributes* (which fields are indexed).

4.1 Type 2 Systems with Realistic Weights

It should first be observed that we have no control over the design of the final database used by a Type 2 system. Hence, when we speak about realistic weights (or easy updating, later), we intend the behavior of the first subsystem, that involved in the selection of an information source. As noted in Section 2.2, realistic weights are possible in components such as a rule base (R), a hypertext database (H), and a page-based database (P). Systems whose first subsystem is built from only one of these components may be called 2R, $2H_s$, $2H_{si}$, $2P_s$, and $2P_{si}$ systems, where the subscript i indicates that the database has information about the indexing attributes of the information sources to be recommended. Updating will not necessarily be easy in systems like this.

These first Type 2 systems function essentially like Type 1R, $1H_s$, and $1P_s$ systems that produce an intermediate answer in the form of one or more information sources (realistic weights now make sense). The system then goes to the database comprising each information source and matches the question attributes against that database in order to determine the final answer. So, two selection operations (or matches) are used: the first to determine a set of information sources and the second to match the question attributes against the databases comprising these sources to obtain the final answer.

Suppose we have a Type 2R system. As an example of its operation, consider a transaction in which a user wants to find biographical information on Linus Pauling, a chemist. That is, the question attributes are: *question-type* = biographical; *personal-name* = Pauling, Linus; *geographical-area* = U.S.A.; and *subject* = chemistry. Suppose the rule base contains a rule saying:

> IF the question-type is biographical, and the geographical-area is U.S.A.,
> THEN Who's Who in America may be useful.

Clearly, the question attributes will match this rule, and *Who's Who in America* will be among the tools recommended. Suppose the system has access to this tool as an online database (for example, in CD-ROM form). Then the question attributes will now be matched against the database; effectively, this means that the personal name will be matched against the database. If a match occurs, then it will be a final answer. The same secondary matching will be carried out with any other recommended tools.

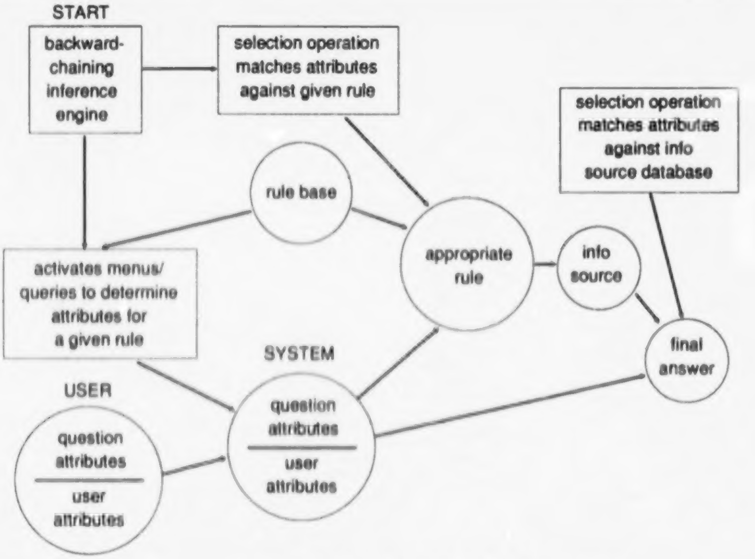

START
backward-chaining inference engine
selection operation matches attributes against given rule
selection operation matches attributes against info source database
rule base
appropriate rule
info source
activates menus/ queries to determine attributes for a given rule
final answer
SYSTEM
USER
question attributes
user attributes
question attributes
user attributes

Figure 5. A type 2R model with a backward-chaining rule base

As with Type 1R systems, the determination of the question attributes may be either by backward- or forward-chaining on the first matching operation. Fine tuning of the attributes may be done if the results of the second matching operation are unsatisfactory.

In Type $2H_s$, $2H_{si}$, $2P_s$, and $2P_{si}$ systems, the internal operation will differ from that of Type 2R systems. But all five cases share the following characteristics:

- the question attributes are somehow determined from the user;
- they are matched against either the IF clauses of rules in a rule base, or the descriptors in a page-based database or a hypertext database;
- the answer is in the THEN clause of the rules, or the answer field in the database;
- a given answer (title of an information source) may appear in more than one place in the rule base or database, hence allowing for realistic weights, but also allowing problems in updating the answer; and

- the question attributes are then matched against the final database (the information source itself) to obtain the final answer.

It should be noted that Type 2R systems may or may not allow precision or indexing attributes, depending on how the rule base has been designed. Hence a Type 2R system may have the system features of any of the four hypertext or page-based systems mentioned above.

Systems that know about indexing attributes (Types $2H_{si}$ and $2P_{si}$) will have an additional match. In our example, the record also has *indexed-field* = personal-name. After the first match that determines the info-source-title (the intermediate answer) for the record, there is a second match that is *not* a selection operation: it merely verifies that one of the indexed fields in the selected tool corresponds to one of the question attributes for which a value is known. Here there is a match, since the personal-name field in *Who's Who in America* is an indexed field.

Special care needs to be taken in the construction of a $2P_{si}$ system. Since every page can have only one set of indexing attributes attached to it, all sources listed on a given page must have the same set of indexing attributes. This problem does not arise with $2H_{si}$ systems, since we have precise labeling of information on hypertext pages, and can therefore assign individual indexing attributes to each label on a hypertext page.

4.2 Type 2 Systems with Easy Updating

As in Section 4.1, when we speak about easy updating, we intend the behavior of the first subsystem, that involved in the selection of an information source. As noted in Section 2.2, easy updating of answers is possible in components such as a single-answer-based database (S) or an item-based database (I), but they do not allow realistic weights. There are four kinds of Type 2 systems that may be built from only one of these components. They may be called Type $2S_s$, $2S_{si}$, $2I_s$, $2I_{si}$ systems. (The subscript i indicates that the subsystem has information about the indexing attributes of the information sources that it will recommend.)

These second Type 2 systems function essentially like Type $1S_s$ and $1I_s$ systems that produce an intermediate answer in the form of one or more information sources. The system then goes to the database comprising each information source and matches the question attributes against that database in order to determine the final answer. So two selection operations (or matches) are used: the first to determine a set of information sources and the second to match the question attributes against the databases comprising these sources to obtain the final answer.

Suppose we have a $2S_s$ system. As an example of its operation, consider the same question as in Section 4.1, namely a request for biographical information on Linus Pauling, a U.S. chemist. Suppose further that the database of information sources contains a record including the following fields and values:

- question-type = biographical
- geographical-area = U.S.A.
- subject = chemistry
- info-source-title = "The following information source may be useful: *Who's Who in America.*"

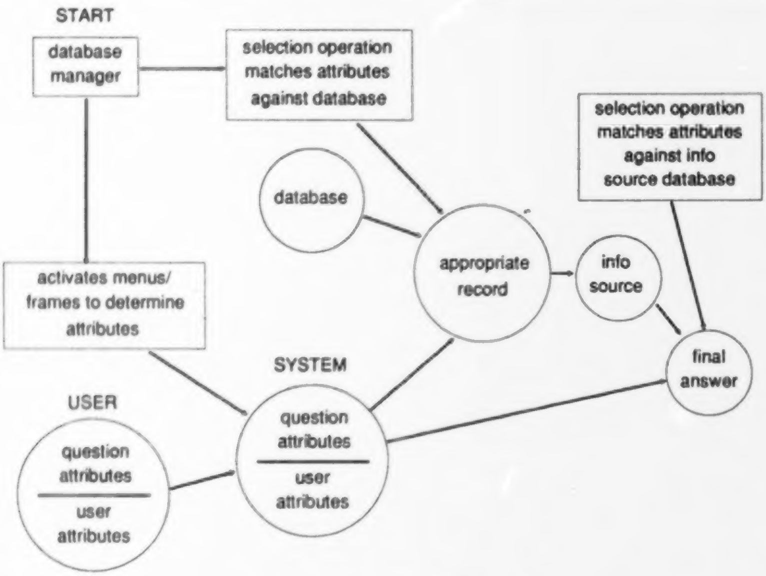

START
database manager
selection operation matches attributes against database
selection operation matches attributes against info source database
database
appropriate record
info source
activates menus/ frames to determine attributes
final answer
SYSTEM
USER
question attributes
user attributes
question attributes
user attributes

Figure 6. A type $2S_s$ or $2I_s$ model

The question attributes gathered through the reference interview will match this record. Unfortunately, a Type $2S_s$ system cannot link effectively with an information source in the form of an external electronic database, since the system is unable to tell which part of the info-source-title is actually the title and which is additional text. The system is a partial implementation of a Type 2 model since the second match must be left to the user of the system.

Suppose, instead, that we have a Type $2I_{si}$ system. As an example of its operation, consider the same question as above. But suppose that the info-source-title in the record has the value: ***Who's Who in America*** and that the record also has ***indexed-field*** = personal-name. After the first match, which determines the info-source-title (the intermediate answer) for the record, there is a second match that is not a selection operation; it merely verifies that one of the indexed fields in the selected tool corresponds to one of the question attributes for which a value is known. Here there is a match, since the personal-name field in ***Who's Who in America*** is an indexed field.

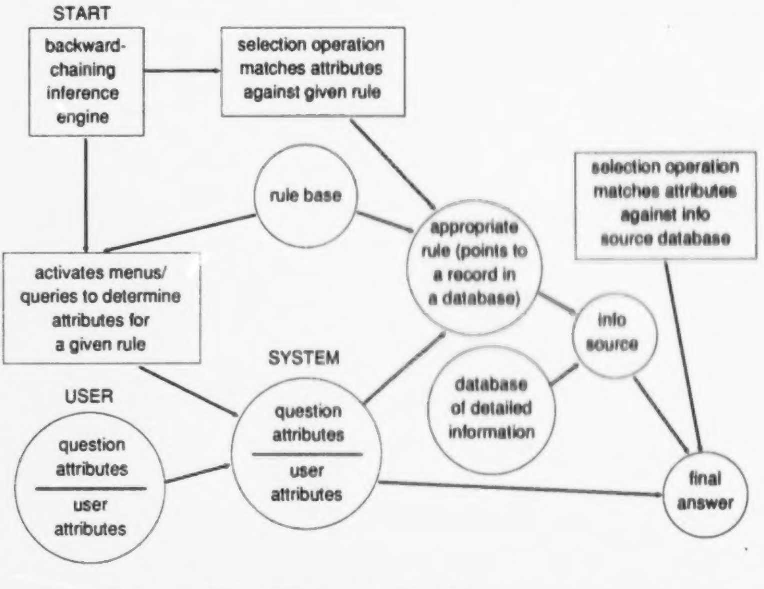

START
backward-chaining inference engine
selection operation matches attributes against given rule
rule base
selection operation matches attributes against info source database
appropriate rule (points to a record in a database)
activates menus/ queries to determine attributes for a given rule
info source
SYSTEM
database of detailed information
USER
question attributes
user attributes
question attributes
user attributes
final answer

Figure 7. A type 2RS or 2RI model with a backward-chaining rule base

Two points should be noted. First, the system has precision in identifying the information source (since no additional text is present); hence it is possible to have a *full implementation* in which the system links to an external database. Second, the system knows about the indexing attributes of the external information sources; hence the expert system retains control over which index to search. The recommendation will be to search the personal-name field of the latter tool. This is done in the final match.

The behavior of the two remaining types, $2S_{si}$ and $2I_s$, may be deduced from the above descriptions of $2S_s$ and $2I_{si}$ systems. Although all four types differ in questions of precision and indexing attributes, each of them allows easy updating but not realistic weights.

4.3 Type 2 Systems with Realistic Weights and Easy Updating

In Section 4.1, we considered Type 2 systems with realistic weights in the first subsystem but not easy updating; some had precision and indexing attributes and some did not. In Section 4.2, we considered Type 2 systems with easy updating in the first subsystem but not realistic weights; again, some had precision and indexing attributes, some did not. In both those sections, we looked at systems with the first subsystem built from one component. As in Section 3.3, it is possible to construct multicomponent subsystems that have *both* realistic weights and easy updating, but not subsystems that have *neither* of these features. Such a multicomponent subsystem matches question attributes against a rule base (or H database) but does not store the final answers in the rules (or records of the H databases). Instead, each rule (or record) points to one or more entries in an I or S database of final answers, which is consulted in constructing the display used to answer the question.

So, as in Section 3.3, the first component must be R or H in order to provide realistic weights; and because we need precision in our link to the next component, the R must be designed to allow that (the H always allows it). The second component must be S or I in order to allow easy updating; it may or may not have indexing attributes. Hence, we have the following possibilities: 2RS, 2HS, $2RS_i$, $2HS_i$, 2RI, 2HI, $2RI_i$, and $2HI_i$.

Suppose we have a 2RS system. As an example of its operation, consider the same kind of question as before, namely, a transaction in which a user wants to find biographical information on Linus Pauling. Suppose, further, that the rule base (the first component) contains the rule

> IF the question-type is biographical, and the geographical-area is U.S.A., THEN go to record Y in the database of single answers for some information sources that may be useful.

The question attributes match this rule, and record Y in the S database gives an answer something like: "The following tool may be useful: *Who's Who in America*." In this case, the system has realistic weights and easy updating but neither precision nor indexing attributes. Consequently, the system is limited to a partial implementation (it cannot perform an online link to an external database); furthermore, it cannot recommend which index of the database to search.

Suppose we have a $2HI_i$ system. As an example of its operation, consider the same kind of question as before. Suppose further that the H component has a page indicating that for biographical information covering the U.S.A., the user should press a "button" leading to record Y in an item-based database (with indexing attributes) of information sources. The question attributes match this button on the page, and record Y gives the answer: *Who's Who in America*. In addition, the question attributes are matched against the indexing attributes of record Y, and the system recommends using the personal-name index. In this case, the system has realistic weights and easy updating, as well as precision (since the second component is item-based) and indexing attributes. Consequently, a full implementation is possible. In addition, the system can retain some control over the final match on the external database, namely the decision of which index to search.

SYSTEM TYPES	REALISTIC WEIGHTS	EASY UPDATING	PRECISION ON TOOLS	INDEXING ATTRIBUTES
2Ss	N	Y	N	N
2Ssi	N	Y	N	Y
2Is	N	Y	Y	N
2Isi	N	Y	Y	Y
2R 2Ps	Y	N	N	N
2R 2Psi	Y	N	N	Y
2R 2Hs	Y	N	Y	N
2R 2Hsi	Y	N	Y	Y
2RS 2HS	Y	Y	N	N
2RSi 2HSi	Y	Y	N	Y
2RI 2HI	Y	Y	Y	N
2RIi 2HIi	Y	Y	Y	Y

Figure 8. Comparison of Type 2 models

4.4 Comparison of Type 2 Models

We are now in a position to compare the various Type 2 models in terms of the four system features: (1) the assignment of realistic weights, (2) easy updating, (3) precision in identifying information sources (and the consequent possibility of full implementations), and (4) knowledge about indexing attributes.

Of the sixteen possible combinations of these four features, only twelve are feasible, since in Section 2.2 we eliminated the possibility of a system with neither realistic weight nor easy updating. Even if it were possible to construct systems like that, it would be hard to justify doing so, since these systems would lack the two most desirable features of the four.

Each Type 2 model has been assigned to one of these twelve categories. It should be noted that the presence of a Y means that the feature is *allowed*, not that the feature is *required;* the presence of an N means that the feature is not allowed. For example, those models in Figure 8 with Y for realistic weights certainly allow the system to have realistic weights, but the system designer is not required to implement this by actually setting up the records or rules so that they have weights attached. Those models with N for realistic weights cannot have realistic weights at all.

4.5 Implementations of Type 2 Models

Because the literature describing implementations does not always give details of system design, it has been difficult to classify some implementations. For example, some systems classified as using I databases may actually use S databases or even P databases. And even the use of a hypertext design tool does not necessarily guarantee that the resulting system uses a hypertext database as we have defined it.

Finally, it should be noted that most implementations, with the exception of item 6 below, appear to be *partial* implementations, that is, they do not have direct access to external electronic databases.

1. An early system, REFSEARCH, was constructed by a group of researchers (included among them was Howard White, now at Drexel) at the University of California, Berkeley (Meredith, 1971). The system has detailed classifications and scopes for the database of reference tools and is a Type $2I_s$ system.
2. Both REFLES (Bivins & Palmer, 1980) and REFLINK (Bivins & Eriksson, 1982), give, among other things, brief instructions for handling unusual searches (e.g., patents), evidently mentioning the information sources for searching patents. Both systems have subject access to the page-based database. REFLINK also has a hierarchical set of menus. So, both REFLES and REFLINK are of Type $2P_s$ (as well as Type $1P_s$).
3. The Reference and Information Station (Purdue University Undergraduate Library), which was mentioned under Type $1P_s$ models, gives menu access to pages on reference tools that might help in preliminary work in a subject area (Smith & Hutton, 1984; Smith, D., 1989). Hence, this system is of Type $2P_s$ (and of Type

$1P_s$). It also features an electronic suggestion box as well as a statistical subroutine for collecting data on use of the system.

4. The Online Reference System (ORS) works by annotating selected records in the automated circulation system (Chisman & Treat, 1984). It allows direct subject access and menu access by type of reference work (and specific class assignment, too) to annotated records in the automated circulation system. Hence, this item-based system is of Type $2I_s$.
5. The Information Function (IF) at Carnegie-Mellon University (Diskin & Michalak, 1985) provides (within the online catalog) menu access to online versions of library publications. This page-based system is of Type $2P_s$ (and Type $1P_s$).
6. The National Agricultural Library (NAL), Beltsville, Maryland, developed a small "demonstration" expert system called ANSWERMAN (Waters, 1986) to help library clients find answers to ready-reference questions. It uses a series of menus to narrow down the subject of the question and the type of tool needed (directory, encyclopedia, atlas, etc.). A set of choices from these menus activates a rule that points to a record in a bibliographic database giving a brief bibliographic description, call number, and, occasionally, an exact page reference. We shall consider other features of ANSWERMAN later in Sections 6.1 and 6.6. Using the same expert system shell, NAL has also developed AquaRef, an expert system for a specialized field, aquaculture (Hanfman, 1989). These item-based systems are both of Type 2RI.
7. POINTER is a system developed by Karen F. Smith at the Library of the State University of New York, Buffalo, for aiding library clients in locating U.S. federal government publications (Smith, K. F., 1986, 1989). POINTER points to reference tools that will help the user find both specific publications and publications on a particular subject. It uses menus to narrow down the type of question being asked. This page-based, menu-driven system is of Type $2P_s$.
8. ORA (Parrott, 1986), developed at the University of Waterloo, allows menu and subject access to pages listing information sources. Hence, this system is of Type $2P_s$ (and Type $1P_s$).
9. PLEXUS is a system developed at the Central Information Service, University of London, as a referral tool for use in public libraries (Vickery & Brooks, 1987; Vickery et al., 1987). It is an ambitious creation including knowledge about the reference process, information retrieval, certain subject areas, reference sources, and library users. The system uses *rules, frames,* and *semantic networks.* It employs *user modeling* and a sophisticated blend of *natural language processing, frames,* and *semantic networks* for handling the reference interview for subject queries. Although the subject

domain is limited to gardening for the prototype phase, it is intended to be broadened in the second phase of development.

PLEXUS uses a database of information sources of four types: publications, organizations, databases, and experts. Hence PLEXUS is an item-based system which begins with an elaborate system for determining the question attributes. Next, rules are used for transforming the question attributes into a concept map, which is then matched against the database of information sources. These rules correspond to various search formulation tactics and term tactics articulated by Marcia Bates (1979). Since we are actually doing a match between the question attributes (in concept-map form) and our item-based database of information sources, PLEXUS must be a Type $2I_s$ system. Incidentally, these types of Bates tactics are also used to modify the concept map if the search misfunctions in some way, e.g., too many or too few hits. Although PLEXUS does not appear to use the Bates WEIGH tactic, it does use user modeling (see below under Section 6.5). In conclusion, PLEXUS is of Type $2I_s$ with several Bates variants.

10. The Information Machine (Fadell & Myers, 1989) is a page-based, menu-driven system that includes pages listing specific information sources. Hence the system is of Type $2P_s$ (and $1P_s$).
11. The Technical Writing Assistant uses a natural language expert system to determine the question attributes, which are then matched against a database of information sources (Butkovitch et al., 1989). This item-based system is of Type $2I_s$.
12. A prototype system developed by Trautman and von Flittner (1989) uses a database of online databases classified by nine attributes. It has several submodules that, among other things, determine the viewpoint (subject), construct a user model, transform the question attributes to a Boolean search, and rank the output. This item-based system is of Type $2I_s$.
13. The Apple Library Tour (Ertel & Oros, 1989) uses a hypertext database mainly to provide directional information. It appears, however, to include hypertext pages referring to information sources as well; if that is the case, then it is of Type $2H_s$ (as well as $1H_s$).
14. Paul Carnahan (1989) shows how to construct a hypertext system that uses Boolean searching of keywords to find reference tools. The system allows the search to be limited further by material type. The *search card* is essentially an interface program that carries out the reference interview and subsequent match against the *database stack* (containing information on the various reference tools). The database stack seems to consist of hypertext pages each of which is restricted to one tool only. Hence the possibility of realistic weights cannot be implemented; on the other hand, easy updating is possible.

Thus, the design of this hypertext database forces it to behave like an item-based or single-answer-based database. Although superficially the system seems to be of Type 2H$_s$, it is probably more correctly classified as Type 2I$_s$ or 2S$_s$.

5. SYSTEMS WITH THREE SELECTION OPERATIONS

Why bother going beyond Type 2 systems? Type 2 systems allow us to model the fact that librarians use *specific strategy* (prescribing the use of specific information sources). But, reference librarians sometimes also use *general strategy* (prescribing the use of categories of information sources); this is acknowledged in another model of Katz (discussed below), and is the basis for Type 3 systems. General strategy, like specific strategy, forms an intermediate answer, and therefore may not always be part of the explicit prescription to the user. But, even if not explicitly stated, general strategy has these advantages:

1. It serves to eliminate from consideration those categories of tools that it does not recommend. This is useful, since many tools may match the usual scope attributes (subject area, geographical area, etc.) but may actually be of very little use in answering the type of question being considered. This is a practical advantage that may be of use in any implementation.
2. It represents a classification of our specific strategies, and therefore allows us to organize our reference knowledge better. This may be useful to the people formulating the reference knowledge; it does not help the user directly.
3. Some inference engines allow explanations (a kind of limited instructional feature). Including knowledge about general strategy allows *explanations of explanations* of specific strategy by indicating that a specific strategy is an instance of a particular general strategy.
4. Intelligent CAI systems (Intelligent Tutoring Systems), are a more comprehensive instructional approach. Including knowledge about general strategy in such systems allows them to teach it. In fact, an ICAI system virtually *requires* the teaching of general strategy, since people find it easier to learn specific strategy if it is presented as a consequence of general strategy (Clancey & Letsinger, 1981).

Type 3 models are derived from *Katz's Translation Device Model* (1982, pp. 76-81). The basic idea here is that:

- after data gathering, a useful type (or types) of information source is determined;

- the latter data plus the data gathered are then used to determine a particular information source (or sources) that may contain the desired information; and
- that source (or sources) is then consulted.

As with the Type 2 models, the Type 3 models apply to directional, ready-reference, and substantive transactions.

Type 2 systems had only one match whose design we could control; hence we needed to consider all combinations of only four features for combinatorial completeness. Type 3 systems, however, have two matches whose design may be controlled. We therefore must consider all combinations of eight features for combinatorial completeness, for a total of 256 possible combinations. But we can reduce this number considerably by reasoning. The eight features are:

1. realistic weighting (first component)
2. easy updating (first component)
3. precision (first component)
4. indexing attributes (first component)
5. realistic weighting (second component)
6. easy updating (second component)
7. precision (second component)
8. indexing attributes (second component)

In the following, we shall examine five of the above eight features and show that the values of none of them may usefully be varied. The arguments will demonstrate either that a given feature must always have a particular value (e.g., positive), or that a given feature is of no interest.

The third feature, precision of labeling in the first component, must always have a *positive* value. The situation here is that we have come to a record or rule that recommends a certain type or types of information sources. But if the answer field does not label the type or types precisely, that is, if it precedes or follows the type with additional data, then it cannot pass the types on to the component of the system in which particular information sources are determined. (A human could, of course, parse this information out, but we assume that the system cannot: that is, it considers the answer field simply a meaningless jumble of characters.) Hence, in a Type 3 system, precision of labeling in the first component must always have a positive value.

The fourth feature, indexing attributes in the first component, is unnecessary. Indexing attributes are important so that we can determine how a particular tool is to be used. Although we could include a default value of this feature for a *class* of tools (for example, "A trade directory generally has an index by manufacturer name"), there is always the possibility that that default may be overruled by the indexing attributes for a particular tool in that class. Furthermore, if we do want indexing attributes, then we must certainly include specific indexing attributes, exactly because we cannot count on a tool conforming to type. So, if the only purpose of this default is to specify the indexing attributes of particular tools, it is superfluous. But what other purpose could it possibly serve? We therefore overlook indexing attributes for the first component.

The fifth feature, realistic weights for the second component, must always be *negative*. The argument here is rather more elaborate. First, by definition, in a Type 3 system: *The type of information source must be determined first, as a necessary preliminary to determining second the particular information sources.* Hence, the second match must be not only on the question attributes, but also on the *type* of information source. This definition implies that, even if a model deduced a particular information source using two matches like this, it would not be a true Type 3 model if one could find another model that would do that in a single match.

Suppose, for the moment, that a Type 3 system *could* have realistic weights for the second component; that is, suppose that realistic weights of the second component could be *positive*. (On this assumption, we proceed to demonstrate a contradiction.) Let the first component be equivalent to any first component in a Type 2 system. But, let the second component be equivalent to only a Type 2 first component with realistic weights (or realistic weights and easy updating).

Given the question attributes, the match in the first component determines the type of information source. The question attributes and type are then matched in the second component to get a particular source. Now, as mentioned above, the generally applicable practical advantage to calculating the type of information source is that it acts to eliminate possibilities in the second match. For example, for a biographical question, if we determined that appropriate types of tools include only biographical dictionaries, general encyclopedias, etc., then the second match will exclude all tools that do not satisfy these types, even though they satisfy all the scope attributes, like geographical area.

But if the second component has realistic weights, as assumed, then it also has eliminative capabilities, as established in Section 2.2. Hence, we can use the eliminative capabilities of the second component to accomplish what the calculation of type did; hence type is unnecessary. This contradicts the definition of Type 3. Therefore, a true Type 3 system cannot have realistic weights for the second component.

The sixth feature, easy updating for the second component, must always be positive. This follows from (1) the previous result, that realistic weights (second component) must always be negative, and (2) we cannot have both realistic weights and easy updating negative in the same component (Section 2.2).

Let us now return to the first feature, realistic weighting for the first component. We shall now argue that it should always be *positive*. Suppose that it were negative; that is, suppose that the first component did not have realistic weighting. Then the first component would be either an I or S database. But we concluded above that precision must be turned on in the first match, so the first component would be an I database. Now, we deduced above that a true Type 3 system cannot have realistic weights for the second component. Thus, the second component would be an I or S database.

So, if realistic weights are turned off, then the system matches the question attributes against an I to get a type of information source. Then it matches the question attributes plus the type of information source against an I or S database to get a particular information source. But recall that in Section 3.3 we argued that a two-component system in which both components allow easy updating but not realistic weights is equivalent to a one-component system that allows easy updating but not realistic weights. So, on our assumption that realistic weighting for the first component is turned off, our system collapses to a Type 2 system. Thus realistic weighting must be positive in the first match in a true Type 3 system.

Finally, there are only three features that can be varied combinatorially (for a total of eight possible combinations):

1. Easy updating for first component
2. Precision for the second component
3. Indexing attributes for the second component

Of the remaining five features:

4. Realistic weights for first component must always be Y
5. Precision for first component must always be Y
6. Indexing attributes for first components are irrelevant
7. Realistic weights for second component must always be N
8. Easy updating for second component must always be Y

5.1 Characteristics of Type 3 Systems

In the previous section, we established five constraints on Type 3 systems; of these, four involved fixing the values of features. We now enumerate some implications of some of those fixes:

- By (4) above, the first component must be one of: R, P_s, P_{si}, H_s, H_{si}, RS, HS, RS_i, HS_i, RI, HI, RI_i, or HI_i.
- By (5) above, we must have precision in the first component, so we are left with first components of: R, H_s, H_{si}, RI, HI, RI_i, or HI_i.

- By (7) above, the second component must be one of: S_s, S_{si}, I_s, or I_{si}.
- By (8) above, the second component must have easy updating, but that applies to all four possibilities just found.

So we are left with $7 \times 4 = 28$ possibilities to be distributed over eight categories. Rather than enumerating these possibilities and describing several models, we shall save the enumeration for the comparison chart in the next section. Here we shall simply describe the model (Type $3RI_{si}$) for which an implementation exists, namely REFSIM (Parrott, 1988, 1989).

Suppose we have a Type $3RI_{si}$ model. As an example of its operation, consider a transaction in which a user wants to find biographical information on chemist Linus Pauling. Suppose the rule base contains a rule saying:

> IF the question-type is biographical,
> THEN biographical dictionaries may be a useful type of information source.

Clearly, the question attributes will match this rule, and biographical dictionaries will be among the types of tools recommended. Suppose further that the database of information sources contains a record including the following fields and values:

- type-of-info-source = biographical-dictionary
- geographical-area = U.S.A.
- subject = chemistry
- info-source-title = *Who's Who in America*
- indexed-field = personal-name.

The question attributes gathered through the reference interview, and the deduced type of information source, will match this record. Now, after this first database match, the indexing attributes of *Who's Who in America* are checked against the question attributes. Since the question attributes include a personal name, and the information source is indexed by personal name, the prescription will be to use its personal-name index. Note that if appropriate fields had not been indexed, it would have been necessary to use either the Bates STRETCH variant (Section 6.3) or the Bates SCAFFOLD variant (Section 6.4). The recommendation after the first database can indicate not only useful information sources, but also the techniques by which they should be searched for the given question.

Since this first database is an I database, the system will be able to send its information over to an external database (such as a CD-ROM system) for the final match. (Because REFSIM is a $3RI_{si}$ system, the latter feature is allowed in REFSIM; but it was not implemented.) And, since the first database knows about indexing attributes, the expert system retains control over which index to search in the external database.

If a match occurs, then it will be a final answer. The same secondary matching will be carried out with any other recommended tools found through other matches on the first database.

5.2 Comparison of Type 3 Models

By combining the enumeration considerations at the beginning of Section 5.1 with information from Figure 8 (comparing Type 2 models), we are able to construct Figure 10, which enumerates and compares Type 3 systems.

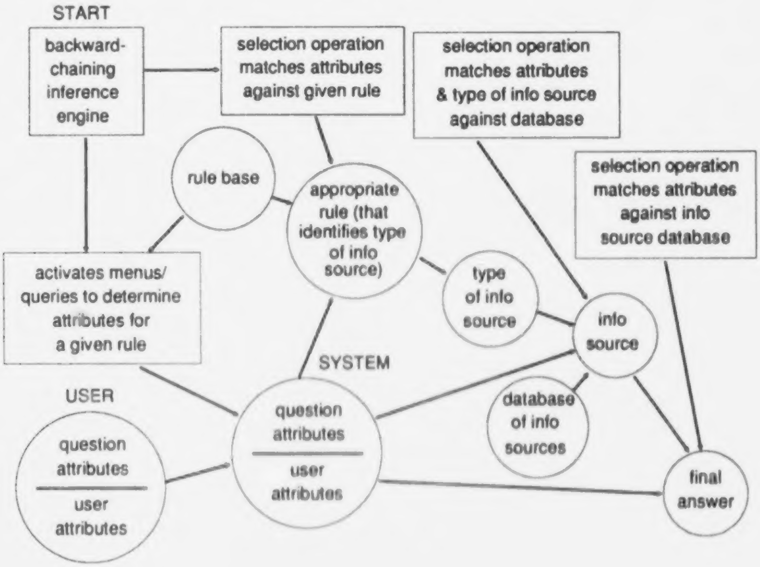

START
backward-chaining inference engine
selection operation matches attributes against given rule
selection operation matches attributes & type of info source against database
selection operation matches attributes against info source database
rule base
appropriate rule (that identifies type of info source)
type of info source
info source
activates menus/ queries to determine attributes for a given rule
SYSTEM
USER
question attributes
user attributes
question attributes
user attributes
database of info sources
final answer

Figure 9. A type 3RS, or 3RI, model with a backward-chaining rule base

6. VARIANTS ON THE BASIC MODELS

In Sections 6.1 to 6.4, we see how certain Bates (1979) search tactics introduce variants in some of the models examined above. The **PATTERN** tactic is deliberately excluded here, since it is so fundamental that it may be thought of as the basic form of many of the operations that are being modified by other Bates tactics in the variants below. In Sections 6.5 and 6.6 we identify other variants.

6.1 Variant 1: WEIGH

With the Bates WEIGH tactic, a weight is assigned to each recommendation to indicate its effectiveness and efficiency in solving the problem. If we allow a system (as opposed to a person) these kinds of weights, we shall call it a WEIGH variant. Now, a glance at the diagrams for our models shows that several important operations may be involved in making any recommendation. Hence, in general, several operations in a model may contribute to the final calculated weight. We may consider a WEIGH variant to arise somehow from modifications (adding weights) to the fundamental operations in a given model.

SYSTEM TYPES		EASY UPDATING (1st comp.)	PRECISION ON TOOLS (2nd comp.)	INDEXING ATTRIBUTES (2nd comp.)
3RSs 3HsiSs	3HsSs	N	N	N
3RSsi 3HsiSsi	3HsSsi	N	N	Y
3RIs 3HsiIs	3HsIs	N	Y	N
3RIsi 3HsiIsi	3HsIsi	N	Y	Y
3RISs 3RIiSs	3HISs 3HIiSs	Y	N	N
3RISsi 3RIiSsi	3HISsi 3HIiSsi	Y	N	Y
3RIIs 3RIiIs	3HIIs 3HIiIs	Y	Y	N
3RIIsi 3RIiIsi	3HIIsi 3HIiIsi	Y	Y	Y

Figure 10. Comparison of type 3 models

It should be noted that Type 2 and 3 models involve consulting sources that may or may not have the desired information; hence, some ranking by likelihood of speedy success would be useful. The WEIGH variant is therefore perfectly natural in these models. It is hard, however, to make a case for using WEIGH variants in Type 1 models, since no intermediate sources are consulted and the final answers are given directly.

Let us now consider the different kinds of operations that have appeared in our Type 2 and 3 models, and how the WEIGH variant might affect them. Those operations include:

1. Commands to perform the reference interview. Weights can be added to the value of each question attribute gathered in the reference interview by asking the user to indicate the importance of each value supplied.
2. Commands to search a system component that allows realistic weights (a rule base or certain kinds of databases) to select an information source or a type of information source (Type 3 only): 2R, $2P_s$, $2P_{si}$, $2H_s$, 2RS, $2HS_i$, etc. or any Type 3 system. Modifications might be of two types:
   (a) Adding weights to the rules or records, to express the likelihood that an information source or type of information source will be useful for the given set of attributes.
   (b) Adding weights to the search commands to express the fact that if we match on a broader or narrower term (see SUPER, SUB, etc. below) than the user really wants, then the likelihood of finding the desired information in an information source that matches is different from what it might otherwise be.
3. Commands to search any other kind of database (*without* realistic weights) to select an information source: $2S_s$, $2S_{si}$, $2I_s$, $2I_{si}$, or any Type 3 system. Modifications might be of two types:
   (a) Here, different weights cannot be specified for different sets of question attributes. The best that can be done is to add one set of weights to each item in the database to express the degree of coverage for that source, given its stated scope.
   (b) Adding weights as in (2b).
4. Commands to consult a database of information sources for additional information (Types 2RS, 2HS, $2RS_i$, 2RI, $2HI_i$, etc.). No modifications to these commands would be reasonable.

Implementations:

- The ANSWERMAN system (Waters, 1986) of the National Agricultural Library, is a Type 2I rule-based system activated by menu choices, and has the capability of attaching weights to its recommendations. Hence, it may also be considered a WEIGH variant of the (2a) type mentioned above. Using the same expert system shell, NAL has also developed AquaRef, an expert system for a specialized field, aquaculture (Hanfman, 1989). All these Type 2I systems use weights of the (2a) variety above.

- The prototype system REFSIM (Parrott, 1988, 1989) of Type $3I_{st}$ allows weights of the (2a) and (3a) varieties above.
- A prototype system developed by Trautman and von Flittner (1989) implements weights of the (1) variety.

6.2 Variant 2: SUPER, SUB, and Other Term Tactics

Sometimes we want more information than we find using the selection operations as described earlier. In general, a system can retrieve additional information by either:

a. allowing matches on reference tools whose scopes are broader than those in the original question attributes, or
b. narrower (if we renegotiate the question), or
c. allowing matches on reference tools whose types are narrower than the type calculated (if we renegotiate the question).

Bates (1979) described term tactics, which help in part of this process. The term tactics move from one search term to a different one; for example, the SUPER tactic moves to a broader term, the SUB tactic moves to a narrower one. The processes in the previous paragraph can be effected by adding (i) *term tactics* just after any of the selection operations, with a control loop to retry the selection operation, and (ii) a *semantic network* of terms on which the term tactics operate. If we allow a system (as opposed to a person) to do this sort of thing, we shall call it a SUPER variant, etc.

We now consider the different types of selection operations and the effects that SUPER, SUB, and other term tactics might have on them:

1. Commands to search a rule base or database to select an information source or a type of information source (Type 3 only).

   If our question attributes match too few (perhaps none) of the IF clauses of any of the rules in the rule base, or the descriptors of any of the records in the database, then we can use technique (a). A SUPER term tactic (operating on a semantic net) could broaden a particular attribute of the question, and then retry the match.

   Consider a biographical question restricted to France, and suppose that there is a rule that says:

   > IF question type is biographical AND geographical scope is Europe, THEN use Z.

   SUPER (operating on a semantic net) could broaden our geographical attribute to Europe and match the rule. And this rule will also be appropriate for a biographical question with geographical scope of France. Rules like this would retrieve additional sources, but these sources might be less effective than sources involving a direct match. Similar considerations hold for a database.

Alternatively, we can use technique (b). Suppose our rule says

> IF question type is biographical AND geographical scope is Paris, THEN use W.

SUB (operating on a semantic net) could narrow our geographical question attribute to Paris and match the rule. But the question attributes would first have to be renegotiated to ensure that the user is interested in Paris. Similar considerations hold for a database.

Technique (c) arises only in Type 3 models, where our question attributes include the calculated type of information source. Notice that *we cannot broaden the type* and then rematch. For example, suppose that the system had first determined that an appropriate type of information source for a telephone number is a telephone directory. If we broadened telephone directory to directory, we might be referred to directories that systematically exclude telephone numbers. But we can narrow the type of information source, for example, to government telephone directory, by using the SUB term tactic. Here it would be necessary to renegotiate the question to see how the type of information source should be narrowed.

2. Commands to consult a database of information sources for additional information (Types 2RS, 2HS, $2RS_i$, 2RI, $2HI_i$, etc.). As with WEIGH, no modification to these commands is reasonable.
3. Commands to match the question attributes against a database of information *within* an information source. The same considerations apply as in cases (1a) and (1b), except that we would generally use a set of term tactics larger than SUPER or SUB. The question attributes need to be renegotiated not only for SUB, but for several other term tactics, including RELATE, NEIGHBOR, TRACE, and FIX.

Implementations

PLEXUS uses the (1a) variety of SUPER when a search statement is being modified because too few information sources were retrieved in the match against the database. This is done by replacing a term by its parent term in BSO, the semantic net used in PLEXUS. PLEXUS also has rules implementing some of the Bates search formulation tactics. For example, after determining the question attributes, PLEXUS uses rules for transforming the question attributes into a concept map, which is to be matched against the database of information sources. A prototype system developed by Trautman and von Flittner (1989) also has some rules like the latter.

REFSIM uses the (1a) variety of SUPER on the REFSIM rule base for choosing a class of information sources if no matches are found.

REFSIM also implements the (1a) variety of SUPER on the database for choosing a specific information source, again, if no matches are found on the given terms.

6.3 Variant 3: STRETCH

With the STRETCH tactic we use an information source for a purpose for which it was not intended. Hence we must first be able to work effectively with single sources, so we must have precision on sources. And we must second have access to information about the intended uses of sources, so we must have all the information about an information source in one place. We must therefore limit ourselves to the following models: $2I_s$, $2I_{si}$, 2RI, 2HI, $2RI_i$, $2HI_i$, and all Type 3 systems with precision on tools in the second component.

Source attributes express intended use. But it is too extreme to allow the ordinary question attributes (subject, etc.) to fail to match the ordinary source attributes. An alternative is to consider failure to match unusual values like the type of information source or the indexing attributes. Various cases are examined below.

1. Match on question attributes, then try but fail on type of information source. This means we must have a Type 3 model. The STRETCH tactic will involve a rule of the form:

   > IF we have a proper match between the ordinary question attributes and the scopes of the information sources, AND there is NOT a match on the type of information source, THEN try the resulting information sources anyway.

2. Match on question attributes, then try but fail on indexing. Since our model must allow matching on indexing attributes, it must be of Type $2I_{si}$, $2RI_i$, $2HI_i$, or of any Type 3 with precision on tools and indexing attributes in the second component. The STRETCH tactic will involve a rule of the form:

   > IF we have found an information source matching the ordinary question attributes, AND IF none of the source fields for which we have input values are indexed in the source, THEN use that information source AND browse over all the data in the information source.

3. Match on question attributes, then try but fail on either indexing or types of information sources. This will require a Type 3 model with precision on tools and indexing attributes in the second component, and will involve broadening the search in the manner of both the (1) and (2) varieties.

REFSIM (Parrott, 1988, 1989) implements the (2) variety of STRETCH variant discussed above.

6.4 Variant 4: SCAFFOLD

The essence of the SCAFFOLD tactic is that we construct an indirect pathway passing through more than one information source in order to reach an information source that will contain the desired information. Hence, we must first be able to work effectively with single sources; we must have precision on sources. And we must second have ready access to all the information about each source, in order to make sure that sources in a pathway have consistent scopes; so we must have all the information about an information source in one place. We must therefore limit ourselves to item-based models. But, to construct the pathway, we must know which fields in our sources are indexed; so we must also have indexing attributes. Hence, as with variety (2) of STRETCH, our model must be Type $2I_{si}$, $2RI_i$, $2HI_i$, or of any Type 3 with precision on tools and indexing attributes in the second component. (Note: A SCAFFOLD variant temporarily forces a 3I system to behave like a 2I system, since it circumvents the command to determine the type of information sources, and operates only on individual sources.) There are at least three types of SCAFFOLDS:

1. A particular tool contains the type of information desired, but is not indexed so that it can accept any of the question attributes as input. A SCAFFOLD tactic here would:
   - assume the final tool and
   - construct the pathway in reverse order so that proper output/input links hold, until we
   - reach a tool that can serve as an initial tool.

   REFSIM implements a two-source version of this type of SCAFFOLD.
2. A particular tool is indexed so that it can accept at least one of the question attributes as input, but it does not contain the type of information desired. A SCAFFOLD tactic here would assume the initial tool, then construct the pathway in forward order so that proper output/input links hold, until we reach a tool that can serve as a final tool. This type of SCAFFOLD is the reverse of the first one.
3. A particular tool contains the type of information desired, and is indexed so that it can accept at least one of the question attributes as input, but the input value does not give a *unique* output (as with "Smith" for a large author index). A SCAFFOLD tactic here would:
   - go to a source that has fewer entries (e.g., one narrower in scope) and has an index for our initial input value;
   - perform a Boolean AND match on the fragmentary input value and other values to be sure we get the correct match (for example, we might try to find all Smiths working in Biochemistry at the University of Leeds); and

- take this more precise value to the final tool.

In this type of SCAFFOLD, unlike the others, we need to be able to do a Boolean-AND match with truncation on the fragmentary value. We can recast these additional requirements as additional Bates tactics. Requiring a Boolean AND here is equivalent to the Bates search formulation tactic, EXHAUST, in which a search is rendered more precise by ANDing all relevant concepts. Requiring truncation in the manner described is equivalent to her term tactic NEIGHBOR, in which we seek additional terms by looking at neighboring terms (in the example given, looking at all terms beginning "Smith"). So, this third SCAFFOLD brings in two more Bates tactics.

Essentially, the SCAFFOLD involves finding ways around the artificial boundaries imposed by the publication process. Insofar as we succeed, we temporarily create a *meta-source* or *imaginary source* that links together the information found in several sources, in order to create the effect of a more powerful source.

6.5 User Modeling

Type 1, 2, and 3 models all include the system's model of the user attributes. This suggests that user modeling is a commonplace feature of reference expert systems. But few reference expert systems have actually implemented it. At the present time, therefore, it is probably better to consider user modeling an optional feature of these various types of models.

Implementations

PLEXUS is one of few systems implementing user modeling. A series of menus is used to determine characteristics of the user. This information is later used, for example, to determine how much explanation of certain tools to give, or to decide how much effort to devote to finding material. REFSIM also provides some support for user modeling.

A prototype system developed by Trautman and von Flittner (1989) also implements user modeling.

6.6 Access to Actual Information Sources

For Type 1 models, the desired information resides inside the expert system. The basic structures of our Type 2 and 3 models, however, explicitly include access to information sources that might contain the desired information. Few current reference expert systems implement this kind of access through an electronic interface; instead, they generally stop at prescribing tools, and leave the consultation of the tools and the final matching to the user. Type 2 or 3 systems lacking this kind of electronic access may be called *partial implementations* of Type 2 or 3 models. It may be noted that a partial implementation of a Type 2 model will have the same basic structure as a Type 1 model, but may have features not necessary or possible in a Type 1 model, such as realistic weights and indexing attributes.

Implementations

The ANSWERMAN system of the National Agricultural Library, mentioned in Section 4.5, is a rule-based system activated by menu choices, and has the capability of functioning as either a consultation system or as a front end to external online databases and CD-ROM reference tools. Using the same expert system shell, NAL has also developed AquaRef, an expert system for a specialized field, aquaculture (Hanfman, 1989). These systems may be the only current reference expert systems that allow this capability. It is safe to predict, however, that this kind of capability will grow considerably in the future.

7. EXTENSIONS TO REFERENCE KNOWLEDGE BASES

7.1 Developing Intelligent Tutoring Systems for Reference

How can a knowledge base on reference practice and theory be used either to instruct library clients or to train reference librarians? Just as expert systems may be used to simulate the professional in the consultational process between client and professional, computer-assisted instruction (CAI) systems may be used to simulate the teacher in the instructional process between teacher and student. But CAI systems are inflexible and inefficient to construct. The situation has been improved with the development of Intelligent Tutoring Systems (ITS), also known as Intelligent Computer-Assisted Instruction (ICAI) systems (Dede, 1986; Peachey & McCalla, 1986). Unlike a CAI system, an ITS typically uses a knowledge base for its subject expertise (as does an expert system) and an additional knowledge base for its teaching expertise.

The subject-expertise knowledge base for a reference ITS is the same as that for an expert system for giving reference advice. So, a single knowledge base could drive both. Since people (unlike machines) find it easier to remember and apply a rule presented as a logical consequence of a strategy, an ITS knowledge base should include heuristic rules giving overall strategy, not just specific strategy (Clancey, 1979). Hence a reference ITS knowledge base will need, for example, rules pointing to classes of information sources, since such rules represent general strategy in choosing information sources. Therefore, reference ITS applications must be Type 3 systems.

Implementations

A prototype ITS system for reference, REFSIM, has been described in some detail in the literature (Parrott, 1988, 1989). A special feature of REFSIM is the simulation of live reference transactions to teach the reference interview and the rationale behind search strategy prescription. REFSIM is a partial implementation of a Type $3I_{si}$ model. That is, it does not allow access to external electronic information sources.

	SURFACE KNOWLEDGE	DEEP KNOWLEDGE
WHEN	usually useful	surface knowledge fails
ARTICULATION BY EXPERTS	easy	usually difficult
DEDUCTIONS	efficient	generally inefficient
DEEP EXPLANATIONS	not possible; can only quote surface knowledge involved	explanations of explanations are possible, by quoting deep knowledge underlying surface knowledge
DEEP ERROR DETECTION	not possible; can detect only errors in understanding surface knowledge	some subtle cognitive errors can be detected, by testing deep knowledge understanding
CREATION OF NEW SURFACE KNOWLEDGE	not possible	possible, but generally time-consuming computations are required

Figure 11. Surface vs deep level knowledge

7.2 Articulation of Deep Reference Knowledge

Deep Knowledge in a subject domain may be thought of as the first principles learned from school or books. This is often the first type of knowledge to be acquired in the professional domain. An expert usually relies on a different type of knowledge, called *surface knowledge* or *heuristic knowledge,* which consists of rules of thumb or short cuts learned from the expert's experience or from experience passed on by mentors (Harmon & King, 1985). A novice normally does not have access to this kind of knowledge.

Deep reference knowledge must correspond to some kind of first principles underlying reference practice. A natural assumption is that some subset of information science underlies reference practice. It is not a great step from the above assumption to the following hypothesis:

> Surface reference knowledge tends to be concerned with the sources of information, but deeper levels of reference knowledge tend to be more concerned with the information itself and the people associated with it.

This approach allows us to establish solid logical links between knowledge in information science and in library science.

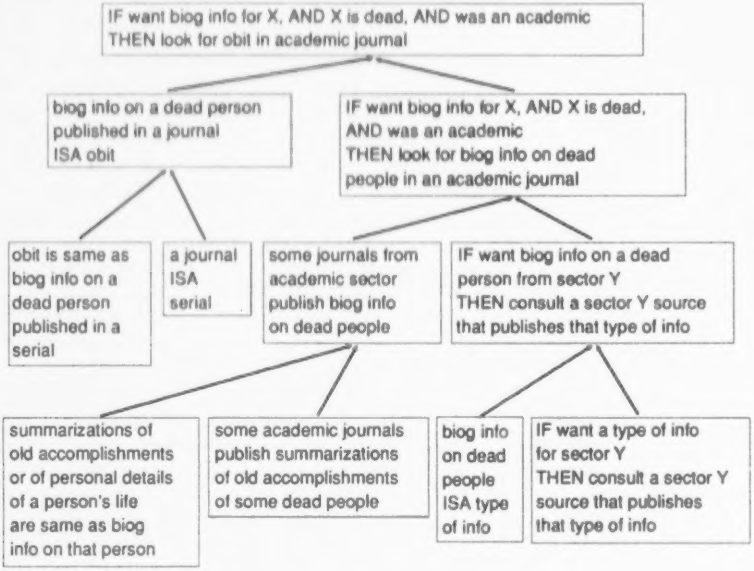

IF want biog info for X, AND X is dead, AND was an academic
THEN look for obit in academic journal
biog info on a dead person published in a journal ISA obit
IF want biog info for X, AND X is dead, AND was an academic THEN look for biog info on dead people in an academic journal
obit is same as biog info on a dead person published in a serial
a journal ISA serial
some journals from academic sector publish biog info on dead people
IF want biog info on a dead person from sector Y THEN consult a sector Y source that publishes that type of info
summarizations of old accomplishments or of personal details of a person's life are same as biog info on that person
some academic journals publish summarizations of old accomplishments of some dead people
biog info on dead people ISA type of info
IF want a type of info for sector Y THEN consult a sector Y source that publishes that type of info

Figure 12. Upper deep structure for a rule

Most of the deepest knowledge deals directly with people and human motivation, and consequently often involves certain *modal* concepts, that is, concepts concerned with what CAN BE, SHOULD BE, IS DESIRED TO BE, WILL BE, etc. This kind of knowledge cannot be expressed well using classical logic, which is concerned rather with what IS. *Modal logic*, an extension of classical logic, is required to represent knowledge like this and to establish the validity of deductions based on this knowledge.

To clarify matters, we shall look at some examples of deep rules from which the following surface rule on biographical information can be deduced:

> IF the type of ready-reference question is biography, the person is dead, and the occupation was academic, THEN consult indexes of academic journals for obituaries.

Some of the upper-level deep knowledge from which this surface rule can be deduced is shown in Figure 12. This rule can be derived from about twenty deep knowledge statements (including those in Figure 12).

8. CONCLUSION: FUTURE PROSPECTS

Many current reference expert systems do not implement some important features of the reference models considered. Of these, user modeling is probably the most critical, and is therefore a promising area for future development. Many current systems might also be improved through the implementation of the Bates WEIGH tactic (e.g., using confidence factors) and the provision of interfaces to external databases. If the last-mentioned facility becomes commonly implemented in reference expert systems over the next few years, then it is only a matter of time before reference expert systems merge with information-retrieval expert systems to form sophisticated front-end systems that can guide a user from one electronic tool to another and give assistance in searching each one of them.

But there is one caveat that must be added. It is widely believed (Walters & Nielsen, 1988) that expert systems in general (including the sort considered in this chapter, as well as those postulated in the last paragraph) have no real future unless the question of the "brittleness" of current expert systems is addressed. Current expert systems are considered to be brittle rather than "robust" since, as they move outside of their areas of expertise, there is a drastic drop in their ability to handle the situation, rather than a graceful degradation. Some researchers believe (Walters & Nielsen, 1988) that providing a knowledge base with deep structure, although a time-consuming process, is a good way of overcoming these limitations of current expert systems.

What is the situation for reference expert systems? The surface structure of reference heuristics and information retrieval heuristics is being well explored in current systems, and the proposed rules seem reasonably consistent with one another. The first principles of information science (which it is reasonable to assume underlie the previously mentioned surface-level heuristics) have been rather less well explored. But the relationship between these two types of knowledge has scarcely been examined at all. This will have to be remedied if we are to make significant progress in creating more intelligent systems. It is this author's conviction that this will indeed happen, and that, moreover, the mapping of these logical links will eventually become as important to the library and information sciences as the mapping of the human genome has become to the medical and biological sciences.

REFERENCES

Bates, M. J. (1979). Information search tactics. *Journal of the American Society for Information Science, 30*(4): 205-214.

Bivins, K. T., & Eriksson, L. (1982). REFLINK: A microcomputer information retrieval and evaluation system. *Information Processing & Management, 18*(3), 111-116.

Bivins, K. T., & Palmer, R. C. (1980). REFLES: An individual microcomputer system for fact retrieval. *Online Review, 4*(4), 357-365.

Butkovich, N. J.; Taylor, K. L.; Dent, S. H.; & Moore, A. S. (1989). An expert system at the reference desk: Impressions from users. *Reference Librarian, 23*, 61-74.

Carnahan, P. A. (1989). Beyond FIND: Boolean searching with HyperCard. *Information Technology and Libraries, 8*, 431-442.

Chisman, J., & Treat, W. (1984). An online reference system. *RQ, 23*(4), 438-445.

Clancey, W. J. (1979). Tutoring rules for guiding a case method dialogue. *International Journal of Man-Machine Studies, 11*(1), 25-49. Also reprinted in D. Sleeman, & J. S. Brown. (Eds.). (1982). *Intelligent tutoring systems* (pp. 201-225). London: Academic Press.

Clancey, W. J., & Letsinger, R. (1981). NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching. In A. Drinan (Ed.), *IJCAI-81: Proceedings of the Seventh International Joint Conference on Artificial Intelligence, 1981* (24-28 August 1981) (pp. 829-836). Vancouver, British Columbia, Canada: University of British Columbia.

Dede, C. (1986). A review and synthesis of recent research in intelligent computer-assisted instruction. *International Journal of Man-Machine Studies, 24*(4), 329-353.

Diskin, G. M., & Michalak, T. J. (1985). Beyond the online catalog: Utilizing the OPAC for library information. *Library Hi Tech, 3*(1), 7-13.

Ertel, M., & Oros, J. (1989). A tour of the stacks: HyperCard for libraries. *Online, 13*(1), 45-53.

Fadell, J., & Myers, J. E. (1989). The information machine: A microcomputer-based reference service. *Reference Librarian, 23*, 75-112.

Hanfman, D. (1989). AquaRef: An expert advisory system for reference support. *Reference Librarian, 23*, 113-133.

Harmon, P., & King, D. (1985). *Expert systems: Artificial intelligence in business.* New York: Wiley.

Katz, W. A. (1982). *Introduction to reference work: Vol. 2. Reference services and reference processes* (4th ed.). New York: McGraw-Hill.

Meredith, J. C. (1971). Machine-assisted approach to general reference materials. *Journal of the American Society for Information Science, 22*(3), 176-186.

Parrott, J. R. (1986). Expert systems for reference work. *Microcomputers for Information Management, 3*(3), 155-171.

Parrott, J. R. (1988). REFSIM: A bimodal knowledge-based reference training and consultation system. *RSR (Reference Services Review), 16*(1-2), 61-68.

Parrott, J. R. (1989). Simulation of the reference process, Part II: REFSIM, an implementation with expert system and ICAI modes. *Reference Librarian, 23*, 153-176.

Parrott, J. R. (1990). Implementation of reference models in expert systems. In R. Aluri & D. E. Riggs (Eds.), *Expert systems in libraries* (pp. 64-75). Norwood, NJ: Ablex.

Peachey, D. R., & McCalla, G. I. (1986). Using planning techniques in intelligent tutoring systems. *International Journal of Man-Machine Studies, 24*(1), 77-98.

Smith, D. E. (1989). Reference expert systems: Humanizing depersonalized service. *Reference Librarian, 23*, 177-190.

Smith, D. E., & Hutton, S. M. (1984). Back at 8:00 AM— Microcomputer library reference support programs. *Collegiate Microcomputer, 2*(4), 289-294.

Smith, K. F. (1986). Robot at the reference desk? *College & Research Libraries, 47*(5), 486-490.

Smith, K. F. (1989). POINTER vs. using government publications: Where's the advantage? *Reference Librarian, 23*, 191-205.

Trautman, R., & von Flittner, S. (1989). An expert system for microcomputers to aid selection of online databases. *Reference Librarian, 23*, 207-238.

Vickery, A., & Brooks, H. M. (1987). PLEXUS—The expert system for referral. *Information Processing & Management, 23*(2), 99-117.

Vickery, A.; Brooks, H.; Robinson, B.; & Vickery, B. (1987). A reference and referral system using expert system techniques. *Journal of Documentation, 43*(1), 1-23.

Walters, J. R., & Nielsen, N. R. (1988). *Crafting knowledge-based systems: Expert systems made easy.* New York: Wiley.

Waters, S. T. (1986). Answerman, the expert information specialist: An expert system for retrieval of information from library reference books. *Information Technology and Libraries, 5*(3), 204-212.

SAMUEL T. WATERS

Associate Director (Retired)
National Agricultural Library
Washington, DC

Expert Systems at the National Agricultural Library: Past, Present, and Future

ABSTRACT

Since 1986, the National Agricultural Library (NAL) has developed four expert advisory systems for ready reference on agricultural topics, and has trained librarians from other institutions who have contributed three other systems. All may be downloaded from the NAL electronic bulletin board. NAL has stimulated development elsewhere of several reference advisors in subjects other than agriculture, and has actively promoted interest in the use of expert systems in libraries. NAL has been responsible for the development of an "intelligent document" in the field of aquaculture, which uses hypertext and contains an expert system. Similar products in several other fields are underway. "Smart" courseware is also being developed for library training. In the future, NAL may explore the use of new artificial intelligence techniques such as neural networks, will increase development of multimedia products and use of multifunctional software, working toward the development of knowledge access and utilization systems in important areas of concern in agriculture.

INTRODUCTION

Several years ago, the National Agricultural Library (NAL) quietly announced that it would distribute a copy of an expert advisory program for ready reference in the field of aquaculture, to anyone who would mail in a floppy disk (Hanfman, 1989, p. 130). Six hundred floppies later, library staff decided to discontinue that offer, loading the expert system on NAL's new electronic bulletin board instead. How did the library become the focus of such interest in a new technology, and where was it going from there?

Background

The National Agricultural Library is one of three national libràries in the United States, the others being the National Library of Medicine (NLM) and the Library of Congress. Unlike NLM, NAL also is responsible for serving as a departmental library. Its staff and budget, small in comparison to those of the other national libraries, must cover the needs of the U.S. Department of Agriculture as well as those of the agricultural community in the United States and worldwide. It is the foremost agricultural library in the world, containing about 2 million items and receiving 26,000 current periodical and other serials from throughout the world.

Networking

Over many years, NAL has built close working relationships with the land-grant universities, most of which still have significant agricultural components. Dealing with a subject of vast scope, from biotechnology to agricultural economics and rural sociology, as well as production agriculture, food and nutrition, and forestry, NAL works with these and other agricultural libraries to meet the needs of a wide array of users of information related to these topics.

Budget and staff limitations have forced NAL to adopt and emphasize certain strategies in attempting to meet its responsibilities. This author shall try to point out some of these emphases in describing the evolution of advanced information technology projects at NAL, and use them in forecasting what the future holds for NAL and libraries in general.

THE BEGINNING OF EXPERT SYSTEMS AT NAL

Several years before the expert system mail-order inundation occurred, some members of the NAL staff decided that the videotape being used for introducing the public to the library was out of date. It would be expensive to redo the whole tape, and there were no funds in the budget for that purpose. Furthermore, only certain parts of the tape were obsolete: much was still viable.

As an inexpensive alternative, we wondered whether we could convert the tape to videodisk and "repurpose" it so that the obsolete material could be hidden. At the same time, we wished to enable the user to view only those sections of specific interest. We obtained special-purpose funding, and contracted for the work. Our contractor, Cordatum, used its own proprietary course-authoring software to create a menu-driven package that turned out to be of some use. Butler (1987) notes that the videodisk included "about 200 still pictures of NAL activities, as well as motion video taken from an earlier NAL orientation videotape" (p. 295). These stills were slides often used in previous staff talks about NAL and its database, AGRICOLA (AGRICultural OnLine Access). To exploit the videodisk further, NAL staff designed a brief experimental course in the content, structure, and use of AGRICOLA, with test questions and scores.

We had hoped that the software would be relatively easy to learn, so that one of our staff might be able to refine the package further without additional paid contractor assistance. Unfortunately, this did not turn out to be the case. However, NAL had gained valuable experience with videodisk courseware, which was to have a payoff in initiating expert systems.

Educational Thrust

NAL management continued looking for ways to improve the cost-effectiveness of the training we offered librarians in using AGRICOLA. Our trainers offered one-week programs which not only provided in-depth information about the database, but also showed how to access it using either of the two commercial service vendors—BRS and Dialog—that made the database available online. Classes were relatively small, and were expensive to hold at off-site locations. The demand had outstripped NAL's ability to supply training.

The experimental course using the orientation disk described above offered a tantalizing glimpse of the possibilities, but it was far from being a useful substitute for human training. Only a videodisk specifically created for the purpose would be able to do the job.

At that time, NAL staff became aware of a software package (called IMSATT) with great promise for use with videodisks, offering impressive course-authoring and expert system capabilities, including the use of touch screens to facilitate use. The library proposed that the USDA Assistant Secretary for Research and Education make part of his program evaluation fund available to the library to develop a videodisk-based training course for AGRICOLA. Upon approval, NAL concluded a cooperative agreement with the University of Maryland Center for Instructional Development and Evaluation (CIDE) to carry out this task.

Personnel at CIDE had been working with an early hypertext system called The Interactive Encyclopedia System (TIES), since developed into the microcomputer hypertext package called HyperTIES. While numerous NAL staff worked on the course content, CIDE personnel inserted hypertext capability into the courseware along with the IMSATT package. The final product, called AGRICOLearn, is now available on a workstation in the Advanced Technology Demonstration Center at NAL. A second system, with a complete workstation, is lent to universities across the country.

Looking back, several library emphases and strategies can be noted. Perhaps most important is the stress on outreach, and specifically on education, as a major library function. At the same time, one can perceive a strategy of seeking funding for specific information technology projects that could be justified as eventually cost-beneficial to the library, and of considerable use to the USDA and the broader agricultural community. One can also see a willingness to utilize the capabilities of other organizations, whether contractors or land-grant university cooperators.

EXPERT SYSTEMS

Familiarization with courseware—its ability to ask questions, branch off from a decision tree depending on answers, and provide feedback to the user—undoubtedly made the notion of expert systems easier for NAL staff to assimilate. Courseware is used for training, and expert systems for decision making, but they follow similar procedures.

While the AGRICOLearn project was underway, the author of this article learned about an inexpensive, easy-to-learn expert system "shell" called 1st-CLASS. Shell software allowed developers to avoid the effort of learning an artificial intelligence language like LISP or Prolog, since they incorporated their own inference engine and user interface. All one had to do was learn a software package no more difficult than spreadsheet software, and then organize and load relevant information into the knowledge base. 1st-CLASS was an example-based system, rather than the typical rule-based one, but seemed even simpler to learn. Impressed by the potential of expert systems in library work, the author decided to develop a small knowledge base system for ready reference simply to prove that it could be done and that the resulting product could be useful.

Why Reference?

Ready reference work is a library function that readily lends itself to expert "systematization." Many similar questions are asked at reference desks, over and over. Often libraries record these questions, and sometimes they record the answers provided, so the data may be available for use in an expert system. One can select a specialized subject in order to narrow the domain of expertise, and can use simple rules to guide a user to an appropriate information source, or even to the exact information required. Furthermore, expert systems can provide assistance when and where reference librarians are not available. Meanwhile, the volume of reference inquiries seems to be rapidly increasing. Finally, human ready reference is not always accurate. Hernon and McClure (1987) state that:

> the research related to unobtrusive testing is beginning to suggest that, on average, regardless of library type or department, reference staff provide a 50-60% accuracy rate for factual and bibliographic questions....The 55% correct answer fill rate is typically computed on an "easier than average" or "average" difficulty level for the questions. (p. 144)

Answerman: A Proof of Concept

Within a few short months in 1985, a very simple and brief knowledge base dubbed Answerman had been created to demonstrate the software and its capabilities. Other staff members contributed refinements. One wrote a brief program in BASIC that requested and stored user feedback about each system consultation. Another wrote a Crosstalk script that would automatically dial up either Dialog or BRS and log on to AGRICOLA or other databases. Finally, with the help of the Microcomputer Center at the Federal Library and Information Center Committee, Answerman was linked to a bibliographic database on a CD-ROM. This project reflected another important aspect of NAL work with expert systems: "the vision thing." The software selected for the project licensed the free distribution of consultation copies of expert systems developed using the shell. This enabled Waters (1986) to assert that "The ultimate goal should be to enable anyone and everyone to obtain ready access to the entire universe of knowledge" (p. 204). That statement coincided with an appeal for librarians to cooperate in building a universal system, by developing individual expert systems that could be linked.

Aquaculture Expert System

The content of the proof of concept system was not important; it served only to show how a real system would look. Answerman showed that expert advisory systems for ready reference were feasible, but could librarians and information specialists in the real world create their own working, useful systems? Fortunately, in expanding NAL outreach, the Director of the Library had begun to establish a number of new information centers, specialized by subject and/or clientele. Since expert systems worked best in narrow domains of expertise, information centers seemed the best place to begin.

The coordinators of the Aquaculture Information Center (AIC) agreed to try to develop a ready reference system for their subject. They "reviewed patrons' correspondence collected over the past two years and selected the topics most frequently asked. Seven species of animals and two species of plants were chosen for inclusion, as well as a general aquaculture information category" (Hanfman, 1989, p. 117). After the initial choice of a species, the user was offered a menu of different types of information, that then led to a likely reference source. In some cases, such as the names and addresses of trade associations in the field, the actual data were supplied.

With minimal assistance from the developer of Answerman, the AIC coordinators quickly produced AquaRef, the first expert advisory system developed at NAL for distribution to the public. They added enhancements similar to those in the proof of concept, and they arranged to link their system to a CD-ROM containing the Aquatic Sciences and Fisheries (ASFA) database, and to ASFA and AGRICOLA online in Dialog and BRS. Evaluation was provided through a user feedback program written in BASIC and linked to answer screens.

Upon completion and internal review, AquaRef was made available for distribution, generating an overwhelming demand and interest in producing other advisory systems.

Training in Expert System Development

In response to this interest, NAL offered training to a number of its staff members who wanted to learn more about expert systems. Among advisors developed in-house was FNIC-AID (Food and Nutrition Information Center—Artificial Intelligence Demonstration), authored by a member of the center staff. This advisory system included a feature new at NAL. At the start of the consultation, user information was requested. Was the user a researcher, an educator, or a nutritionist? Each answer led to a somewhat different set of questions, or in the case of similar questions, to different reference sources, appropriate to the different types of user needs and skills. The system also included some tabular data from reference sources as answers, rather than just a citation to the name of the reference tool. Another product, resulting from the collaboration of several NAL staff members, was an advisor that led to online search strategies for almost a hundred different popular topics, based initially on "saved searches" for AGRICOLA that had been stored in the Dialog online system. Major topics covered include different geographic regions: soil classes; soils reclamation; insects; crops, plants, and weeds; birds and animals; and nutrition and health (Rafats, 1989).

Still later, the author of this article produced a knowledge base on microcomputer-based expert system tools. Unfortunately, information about products rapidly becomes obsolete in such a rapidly moving field, and without frequent updating the knowledge base becomes less accurate and less useful day by day.

Training for Personnel Outside NAL

While these activities were going on, librarians and information specialists outside NAL expressed a desire to learn how expert systems could be developed. In July 1987, NAL announced a five-day training program to prepare librarians to develop small expert systems in agriculture-related fields. Participants were expected to attend two days of lecture and hands-on laboratory sessions, followed by three days of supervised system development. By the end of the sessions, it was hoped that attendees would have a well-developed prototype plus a plan for its completion.

The course, ending in October 1987, was attended by personnel from AID, the Animal and Plant Health Inspection Service of USDA, BIOSIS, Quaker Oats, NAL, and several university libraries (Swab, 1987). Some of the products created at the class focused on a single library, but several were of more general applicability. One covering organic chemistry, including agrochemicals and pesticides, was developed by Craig A. Robertson of the University of Vermont. Asphalt Forest was the title of one on urban forestry, initiated by Stephanie Chase, then at Colorado State University. It was later completed by Chase and Gilman at the University of Florida. Still another advisor, on Louisiana aquaculture, was prepared by Susan Hocker of Louisiana State University. It covered a number of commercial species, such as redfish, that were not included in AquaRef. Two librarians from the University of Maryland also attended some of these training sessions. Upon returning to their library, they enthusiastically began work on developing expert systems for internal use, a project they found very worthwhile.

Dissemination of Expert Systems

As noted above, NAL had first distributed AquaRef on floppy disks submitted by requestors. Eventually, a different method presented itself. NAL established an electronic bulletin board, known as ALF (Agricultural Library Forum), accessible around the clock seven days a week at (301) 344-8510 and 8511 (Pisa, 1988, p. 6). In addition to messages and bulletins, computerized text and software may be downloaded from ALF. Currently, four NAL advisors and three university-prepared advisors are being distributed by this method. The bulletin board has just become available in the United States through a toll-free number, 1-(800)-345-5785. While this dissemination technique reduces the drain on NAL resources, it has disadvantages for market research (tracking who is using which files), and makes it harder to obtain feedback for product improvement.

Informal Advice and Assistance to Other Organizations

Other organizations began to hear about NAL expert system activities through publications, announcements, and demonstrations in the NAL Advanced Technology Demonstration Center, and by word of mouth.

One of the early visits to NAL came from the staff of Goucher College in Baltimore. Librarians there were considering the development of a biographical reference expert system, and their computer advisors were impressed by the ease of use of the 1st-CLASS software. Larry Bielawski was director of the Decker Center for Information Technology at Goucher College, and Robert Lewand was professor of mathematics and computer science at Goucher. Working with Yvonne Lev of the college library, they produced an impressive knowledge base system using an upgraded version of the 1st-CLASS software, called Fusion (Bielawski & Lewand, 1988, p. 63).

After Karen Patrias, from the National Library of Medicine, returned to her office from a visit to NAL, she and her staff used the same software to begin developing MEDSTATS, an extensive expert system to assist in locating sources of statistical information. Since a single source might be cited many times for covering many different diseases and types of statistics, they sought to avoid needless repetition of the same entry. They did this by entering each bibliographic record only once in a database, coding it for all the different aspects covered. Then they linked the database, with its own special search software, to the expert system.

The Economic Research Service (ERS), a USDA agency, also was influenced by NAL's experience. Jim Horsfield of ERS developed an expert system to answer inquiries about ERS products and services, and to provide referrals to human experts, avoiding the dread scourge of "telephone pass-around." Called "Finders," over 3,500 copies of the microcomputer diskettes have been distributed free at conferences and meetings and to ERS secretaries. The system has been updated several times in the last few years, most recently to 1st-CLASS HT, which has a hypertext capability. The system, like MEDSTATS, is also used as a front end to a database, permitting changes to be made in the database without necessarily requiring changes in the expert system itself (Robb, 1989, p. 15). Among the many institutions directly influenced by NAL expert systems work have been some overseas. For example, Marcus Sahlu of the International Livestock Center for Africa (ILCA) developed an advisor on cattle and a guide to the International Agricultural Research Centers during a training stay at NAL. Directly and indirectly, the NAL work on expert systems had influenced many organizations worldwide.

ELECTRONIC FULL TEXT

Parallel to and related to its work on courseware and expert systems, NAL had been experimenting with the use of laser technology. It had used videodisks in both its orientation and training projects. But it was also exploring ways to store and search the full text of publications in electronic form. Online searching of the full text of selected journals was becoming available, but use of that medium seemed too expensive for all but the cream of library materials. Library management believed that laser technology might offer cost advantages over online use, together with accessibility that would override any other possible benefits of microfilm.

It seemed clear that the information problem was changing from scarcity to overabundance. A search of full-text files might overwhelm a user with hundreds of "hits." Computerized intelligence, whether in the form of expert systems, natural language, and/or some other approaches, would be necessary to ameliorate this problem.

NAL learned that a firm named LaserData, a systems integrator, was using BRS software to experiment with the use of videodisks to store text. This was done by converting digital information to ride an analog signal on the videodisk, which could also store and reproduce images in the analog mode. Obtaining USDA program evaluation funds to initiate a full-text project, NAL decided to use the text (and illustrations) of the *Pork Industry Handbook*. (It was only a coincidence that former pig farmer John Block was then Secretary of Agriculture.) The whole text of the publication had to be rekeyed, an expensive proposition, but the final product contained copies of the illustrations linked to the sections of text to which they were related, and every significant word could be searched using BRS software.

A second videodisk was then prepared, containing fourteen non-copyrighted publications from the Extension Service. A variety of input methods was used, including intelligent optical character recognition (OCR) by the English firm, Optiram, as well as programming to convert a variety of photocomposition tapes to usable code. This project demonstrated that the latter technique was not the easy, inexpensive alternative to keyboarding that it might have appeared to be at first glance.

Meanwhile, Hernan Otano had been experimenting with full-text input and retrieval at the National Air and Space Museum. The process involved facsimile scanning to create a bit-mapped image of the page and then using OCR software to convert it to ASCII code. The code was automatically indexed, and then stored on a WORM (Write Once Read Many) disk. After viewing demonstrations of that project, NAL staff concluded that it was the way of the future. Obtaining more evaluation funds from the assistant secretary, the director called a number of land-grant university librarians together and persuaded them to contribute $3,000 each to support a cooperative program, the National Agricultural Text Digitizing Project (NATDP). Output for evaluation was to be placed on digital CD-ROMs, seen to be the medium most likely to be used by libraries in preference to videodisk.

With additional contributions from over forty land-grant libraries, and from evaluation funds, the project has resulted in the production of a compact disk containing the text of some sixty important non-copyrighted publications in the field of aquaculture. This disk tested the usability of OCR text with minimal, moderate, and maximum human cleanup. Bit-mapped images of all pages, not just those with illustrations or tables, were placed on the compact disks issued for evaluation. While this process improves the product for preservation, the extra storage required for bit-mapped images of pages without illustrations reduced the number of pages per disk perhaps forty-fold to "only" 6,000 pages (Andre et al., 1989).

A second aquaculture disk, with completely clean text, has been proposed. Meanwhile, a disk prepared by the International Agricultural Research Centers, containing some of their own publications, is being evaluated at NAL, as is a two-disk set of acid rain materials, and a disk containing material from the NAL collection on Agent Orange (Zidar, 1990, p. 2). Just being demonstrated is a disk produced by NAL in cooperation with the USDA Federal Extension Service and State Extension Service offices at VPI and the University of Minnesota. Titled the National CD-ROM Sampler, it contains the text of some 10,000 Extension brochures and publications, and approximately 1,000 graphics. One section on birds offers not only text and graphics, but audio of the birdcalls and songs as well. Evaluation of the disks will cover the different indexing and search software that have been used with them.

A major facet of the NATDP is the evaluation of different software packages used for access to text and images on compact disks. At least one of these packages, Personal Librarian, uses an unconventional approach. The user can conduct a quick keyword or Boolean search and locate a very relevant document. The software can then be used to search the entire file, comparing other documents in the file to the one selected, and ranking their relevance in terms of the number of identical significant words that also appear in the target document. Beyond this, the software can show the most significant words associated with the target and other relevant documents. The user can then build a search strategy with a cluster of associated words and phrases dredged up in this fashion.

Searching by using a cluster of related terms linked together before the search, without having to know or specify all the related terms, is known as *concept searching*. It permits documents highly relevant to a search to be located even though they may not contain the terms specified in the search strategy. It builds knowledge and intelligence into the process of accessing databases, and appears to have a promising future.

One software system that might be described as a concept searcher with a quasi-natural language front end is Tome Searcher. Developed by Tome Associates, it creates a specialized thesaurus which allows a searcher to input a query that is automatically mapped into a search formulation using the linked terms in the thesaurus. Depending on anticipated results, the software can broaden or narrow the search. AWIC, the Animal Welfare Information Center of NAL, has a small contract with Tome Associates to develop a concept-searching gateway for that subject area. A preliminary version with a 6,000 word thesaurus has just been received.

SMART DOCUMENTS AND DOCUMENT COLLECTIONS

NAL expert system activities took an extremely significant turn late in 1988, when a decision was made to create a "smart document" incorporating hypermedia links along with an expert system. Robert Freeman, chief of the Fishery Information Service at the Food and Agriculture Organization of the UN (FAO), had seen a demonstration of AquaRef at NAL. When he decided that a published survey of African aquaculture could be more useful to the staff of FAO in electronic form, he turned to NAL for assistance. Discussions with the head of NAL's Aquaculture Information Center resulted in a decision to use a powerful software package called KnowledgePro. Working with its object-oriented programming language, it was possible to create both hypermedia links and an expert system (Mace, 1989, p. 15). Together, expert systems and hypermedia are greater than the sum of the parts, providing structure and procedural control while offering the opportunity to browse at will.

With funding from FAO, and from the National Oceanic and Atmospheric Administration, the head of the AIC decided on the needed hypermedia linkages, and worked with a USDA expert to create an expert system for African aquaculture, the first nonbibliographic expert system developed at NAL. KnowledgePro programming is considerably more difficult to learn than 1st-CLASS, so that effort was performed under contract by the two Goucher College professors who had previously visited NAL in connection with development of a ready reference advisory system for biographical information.

The resulting product, code named REGIS (REGional Information System), included a map of Africa linking each country to the relevant section of the text, displayed appropriate search strategies, and offered automatic links to the Aquatic Science and Fisheries Abstracts database online or on CD-ROM. The project took only a few months to complete and elicited favorable attention in the computer press. The microcomputer runtime software is being distributed by the National Technical Information Service. The product is now being revised and upgraded to include additional capability, specifically, the power to move directly to appropriate sections of text by searching key words.

Meanwhile, several similar projects have been undertaken. KnowledgePro is being used to create a "smart document" on pesticide applicator training. The Animal Welfare Information Center (AWIC) has also gone beyond bibliographic expert systems. AWIC initiated the development of an expert system with hypertext on the topic of animal anesthesiology, using a veterinarian as the domain expert and the Goucher twosome as knowledge engineers. In this case, they will use the 1st-CLASS HT software earlier used for Finders by ERS. Perhaps these smart document/expert systems presage an important change in library work, a greater emphasis on packaging nonbibliographic information in databases and expert systems.

COMPUTER-ASSISTED INSTRUCTION

The last NAL project to be discussed closes the loop: another education program. CatTutor is a computer-assisted instruction (CAI) tool being developed on a Macintosh. S. E. Thomas and C. V. Weston (1990) state that: "In its initial application, the CAI package focuses on descriptive cataloging of computer software" (p. 2). Relevant segments of cataloging tools (AACR2, the MARC format, and LC rule interpretations) have been scanned and converted into searchable text. Using Hypercard software, these machine-readable files have been linked to each other and to examples in the cataloging tutorial. One notable aspect of the development process was the involvement of an instructional design consultant, which perhaps should become an essential part of the expert system design process as well. Funded initially as a winning proposal in the Apple Library of Tomorrow grant program, the project has since received substantial support from the Council on Library Resources. Two university libraries and some library schools began conducting evaluations of a functioning prototype in Fall 1990.

THE FUTURE

Forecasting the near future should not be too difficult; one simply extrapolates obvious trends. Several important thrusts of NAL activity in the past will probably continue to influence NAL work in the future. First is the emphasis on outreach, exemplified in the dozen-odd specialized information centers already established. Several of these centers have played an important part in the development of advanced information technology. They provide an excellent mechanism for focusing on user needs, allowing a subject specialist to do market research and develop entrepreneurial attitudes towards the application of new technologies to meet those user needs. No doubt many more projects will originate and be nurtured in these centers. Another thrust is cooperation, required since no one organization can do everything that is needed. Some of the projects mentioned above, such as the Text Digitizing Project, will become national programs involving institutions nationwide and even worldwide, requiring stable, long-term funding.

Another important aspect of NAL work in new technologies is the focus on early application, as contrasted to pure research. NAL has been forced to seek the optimal place on the development curve to make its investments, so that it can advance the state of the art relatively quickly and with relatively minimal investment, bringing clear benefits to its users while it transfers the new technologies to its colleague institutions.

Neural Networks

One of the new technologies for which NAL may be awaiting the right development stage is artificial neural networks. With their abilities in pattern recognition, and their ability to learn, neural nets may present excellent opportunities for library applications.

A company called Excalibur markets word pattern recognition software named "Savvy." Savvy can be tuned to find words in text that match misspelled words in search queries and vice-versa. Garbling of words due to imperfect optical character recognition has forced the NATDP to clean up the code manually or face imperfect search recall. If neural net software could do a good job of searching garbled OCR text, cleanup might not be necessary. At this time, Savvy runs on a minicomputer. When it or comparable neural net software is available on a microcomputer, NAL has expressed interest in investigating it further.

Another possible NAL application of neural net software might be in refining current awareness profiles. Users could indicate the degree of relevance to their interests of document selected by their profiles. Neural nets, examining citations and abstracts of the items rated most relevant, might automatically alter the user's search profile, continuing to upgrade it after feedback from subsequent searches. And of course, one should not overlook the fact that microcomputer-based neural net software already has been used for applications like those of expert systems, as well as in conjunction with expert systems, as inductive front ends.

Multifunctional Software

Whether or not neural nets or any other specific type of intelligent software will become useful in library work, it seems unlikely that any single functional type will displace the others. There is a clear trend to the integration of different capabilities within software packages. Much has already been done: some expert systems include two-way links to DBMS and spreadsheet software, along with hypertext and quasi-natural language query modes. It seems clear that user demand will lead to integrated multifunctional software that also includes the use of fuzzy logic, Boolean search, concept search, relevance ranking, neural nets, genetic algorithms, and perhaps even geographic information systems, all linked to full text and motion video.

Why is this likely to happen? Because humans find that no one mode of problem solving, whether analogy or logic based, and no one mode of knowledge representation, whether image or word based, deals best with all situations.

Multimedia Products

A multimedia approach has been characteristic of many NAL applications, from its first orientation videodisk to the latest Extension sampler on CD-ROM. Exploitation of the colorful interfaces used for REGIS may account in part for the lessened interest at NAL in the straightforward, simple decision-tree ready reference advisors. Multiple media are used in smart documents and expert systems for the same reason they are used in courseware: multimedia products attract and instruct by entertaining. The term *edutainment* in current use reflects the marriage of education and entertainment. Before long, librarians may realize they are in the *infotainment* business: offering information products and services that make locating and using knowledge interesting, challenging, stimulating, and motivating.

Some of us may remember the drudgery of plowing through bibliographies printed in monthly issues with quarterly, annual, and quinquennial cumulations, copying relevant citations in longhand. For us, searching a user-friendly CD-ROM and printing out selected citations is FUN! And when we see color-keyed windows and hypermedia links in an electronic encyclopedia, we think we're in Disneyland, or at least the Epcot Center. We need to make library users feel the same way.

Scientific Data

Future products of NAL are likely to be multicontextual as well as multimedia. That is, they are likely to include scientific-numeric databases as well as bibliographic, full-text, audio, and visual files. As NAL becomes more involved in large scientific programs such as those for the plant genome and for global change, it will have to plan to make multiple scientific databases available in a user-friendly fashion. These databases will be huge collections of observations, such as those derived from satellite sensing. Presumably, NAL will participate in the development of end-user workstations with specialized functional software, whether statistical, geographic, or other, as the case may be.

CONCLUSION

End-User Empowerment

User-friendly interfaces, integrated multifunctional software, multimedia products in specialized fields, covering scientific data as well as bibliographic, full-text, and image files: the result, our implicit goal, seems to be empowerment of the end-user. Workstations alone will not be the answer. Perhaps an intermediate step will be acronymed as EUREKAS: End User (Research and Education) Knowledge Access Systems. Beyond that lies the intelligent agent or "Knowbot," the Knowledge Navigator so interestingly limned by Apple's Sculley.

The Long Run

In the short run, we tend to overestimate the amount of change that will occur within our inertia-bound institutions. In the long run, we underestimate it. That may be because incremental change becomes qualitative change, so in the long run change changes into transformation, and the law of unintended consequences has time to make itself felt.

We who have worked to advance the state of the art in library applications of artificial intelligence have a responsibility to try to anticipate some of those unintended consequences of our actions. Moravec (1989) gives the most chilling forecast one can imagine, discussing:

> the intelligent robot, a machine that can think and act as a human, however inhuman it may be in physical or mental detail. Such machines could carry on our cultural evolution, including their own construction and increasingly rapid self-improvement, without us, and without the genes that built us. When that happens, our DNA will find itself out of a job, having lost the evolutionary race to a new kind of competition. (p. 2)

A frightening line disseminated by the Computer Professionals for Social Responsibility goes, "If we're lucky, they may keep some of us as pets." Our creations are gaining intelligence rapidly. Should we be refocusing on wisdom, instead?

REFERENCES

Andre, P. Q.; Eaton, N. L.; & Zidar, J. A. (1989). Scanning and digitizing technology employed in the National Agricultural Text Digitizing Project. In D. L. Blamberg, C. L. Dowling, & C. V. Weston (Eds.), *Proceedings of the Conference on Application of Scanning Methodologies in Libraries* (pp. 61-76). Beltsville, MD: National Agricultural Library.

Bielawski, L., & Lewand, R. (1988). *Expert systems development: Building PC-based applications.* Wellesley, MA: QED Information Sciences.

Butler, R. W. (1987). New technologies: Computerizing the future. In J. J. Crowley (Ed.), *Research for tomorrow: 1986 yearbook of agriculture* (pp. 294-300). Washington, DC: U.S. Department of Agriculture.

Hanfman, D. (1989). AquaRef: An expert advisory system for reference support. *Reference Librarian, 23,* 113-133.

Hernon, P., & McClure, C. R. (1987). *Unobtrusive testing and library reference services.* Norwood, NJ: Ablex.

Mace, S. (1989). REGIS merges two technologies: Hypermedia, expert systems more effective together than apart. *InfoWorld, 11*(12), 15, 18.

Moravec, H. (1989). *Mind children: The future of robot and human intelligence.* Cambridge, MA: Harvard University Press.

Pisa, M. (1988). ALF (Agricultural Library Forum): An electronic bulletin board system. *Agricultural Libraries Information Notes, 14*(10), 1-6.

Rafats, J. (1989). Do you use AGRICOLA saved searches? *Agricultural Libraries Information Notes, 15*(1), 2.

Robb, D. W. (1989). Expert-system shell helps public find USDA data. *Government Computer News, 8*(20), 15.

Swab, J. (1987). NAL's expert systems course off to a good start. *Agricultural Libraries Information Notes, 13*(12), 8-9.

Thomas, S. E., & Weston, C. V. (1990). CatTutor for catalogers at the National Agricultural Library. *Agricultural Libraries Information Notes, 16*(2), 1-5.

Waters, S. T. (1986). Answerman, the expert information specialist: An expert system for retrieval of information from library reference books. *Information Technology and Libraries*, 5(3), 204-212.

Zidar, J. A. (1990). National Agricultural Text Digitizing Project update. *Agricultural Libraries Information Notes*, 16(2), 5-6.

CHRISTINE L. BORGMAN
YOLANDA I. PLUTE

Graduate School of Library and Information Science
University of California, Los Angeles

User Models for Information Systems: Prospects and Problems

ABSTRACT

Expert systems attempt to model multiple aspects of human-computer interaction, including the reasoning of the human expert, the knowledge base, and characteristics and goals of the user. This paper focuses on models of the human user that are held by the system and utilized in interaction, with particular attention to information retrieval applications. User models may be classified along several dimensions, including static vs. dynamic, stated vs. inferred, and short-term vs. long-term models. The choice of the type of model will depend on a number of factors, including frequency of use, the relationship between the user and the system, the scope of the system, and the diversity of the user population. User models are most effective for well-defined tasks, domains, and user characteristics and goals. These user-system aspects tend not to be well defined in most information retrieval applications.

INTRODUCTION

The topic of this conference is artificial intelligence and expert systems in the library setting. The question addressed in this paper is where do "user models" fit in this discussion.

Systems generally are considered "expert" when they have some reasoning ability. The problem domain is usually the object of the reasoning—a knowledge base is built from data about the domain, often combined with knowledge about the relationships among the data drawn from interviews with human experts. User models, in contrast, consist of reasoning about the person who is manipulating personal characteristics that may influence the user of the system, with or without additional data about the problem the user brings to the system. User models may be implemented in combination with other expertise or as the primary expertise in the system.

User models start with some expectation of the knowledge the user brings to the system and about how he or she will interact with the system. The user model allows the system to adapt its interaction style and content to the individual user. User models have several contributions to make to expert systems, according to Karen Sparck Jones (1989): they can increase system *effectiveness*, helping to ensure that the system makes the correct decision. They can serve system *efficiency*, helping to reach the correct decision in an economical way; and they may increase system *acceptability*, in expressing or presenting the results of the system in a way most comprehensible and usable for the individual.

This paper describes and discusses the various types of user models that have been constructed in the context of information systems, and concludes with an analysis of the usefulness, advantages, and disadvantages of implementing user models in information retrieval systems.

RESEARCH ON USER MODELS

One of the purposes for pursuing the construction of user models in information retrieval is to provide systems with "intelligent interfaces," with ease of interaction as the objective. Brooks, Daniels, and Belkin (1985) describe an intelligent interface as "something that stands or mediates between user and knowledge resource in the information system" (p. 191). In an information retrieval setting, the "intelligent interface" can act as a human surrogate, helping the user to clarify and meet his/her information need. Users may not know precisely what is being sought, but can, to some extent, describe the problem that has brought them to the system. In traditional library reference services, the librarian performs this function. Continued development of intelligent interfaces for information retrieval systems offers the possibility of replacing the human intermediary with a system that can perform the query negotiation function traditionally carried out by librarians.

If the system is to replace the human successfully, it must mimic or model the actions of the human intermediary (Brooks et al., 1985).

One thing the system must do is build a model of the user's problem (Brooks, 1986), rather than request a specific statement of the information need, as required in conventional information retrieval systems. It is a function of the intermediary to assist the user in defining the problem precisely. The system must also build a model of the user. As in human-human interaction, when the system and the user engage in a dialogue, each adapts its model of the other in the process, until the problem has been identified satisfactorily. Asking the user for relevance feedback provides the interactive element necessary to arrive at a more accurate problem assessment, since user perceptions of the problem may change in the course of the session. The interactive element allows modification of the problem image within the system in order, finally, to arrive at the appropriate query formulation or information need.

THOMAS, for example, is an information retrieval system that employs a user model and is designed to retrieve bibliographic references in the area of medicine and biochemistry (Oddy, 1977). The objective of the system is to enable users to present a subject term and have the computer carry out a search based on that term, thereby freeing the user of formulating a full search query. The system matches up the term with the item closest to it and presents the user with references. The user reviews the selections and can reject, accept, or make no judgment on them. From the user's response, THOMAS can modify its image of the user's area of interest if necessary, and present alternative selections. In this case, the system models the user to determine the area of interest and expertise, just as a reference librarian might do.

A brief review of the literature illustrates the various definitions of user models and discusses several applications of those models used to provide intelligent interfaces for information retrieval.

TYPES OF USER MODELS

Research in human-computer interaction, and in particular interface design, focuses heavily on the thought processes, or cognitive processes, of the user. De Mey (1977) states that cognitive processes are involved in all information processing activities, and provide the individual with concepts that serve as a model of the individual's world and a way for the individual to organize his/her knowledge. Knowledge of human behavior in information retrieval tasks will be helpful in systems design as well as in user training (Borgman, 1986a).

Various models have been identified that represent the thought processes that occur when two individuals interact. The three major types of models are conceptual, mental, and user models (Borgman, 1986a). These types of models are distinct but complementary.

Conceptual Models

The conceptual model, according to Norman (1983), is the model of a system presented to the user by someone else, such as the designer or a teacher. Halasz and Moran (1982) add that the conceptual model provides the user with information about the underlying structure of the system, giving the user a starting point with which to reason about the system.

A conceptual model of an information retrieval system might be based on a card catalog, for example. A model for a word processing system might be based on a secretary and a filing cabinet. Halasz and Moran (1982) discuss metaphorical vs. abstract conceptual models at length.

Mental Models

The mental model is part of the thought process of the user when interacting with a system. People develop a mental model internally, as opposed to having it presented to them (Norman, 1983). The user's mental model may be based on a conceptual model that has already been presented to him/her or it may be developed independently (Borgman, 1986a). The mental model is how he/she thinks the system is structured and how it functions. Norman (1983) defines the mental model as "what people really have in their heads and what guides their use of things" (p. 12). The user's beliefs about the system will be incorporated into the user's mental model of the system regardless of their accuracy. It often is difficult to ascertain exactly what elements are at work in a mental model, as the user may not be conscious of the presence of a model and cannot clearly articulate the model. The mental model is helpful to the user when first learning to use a system and later can be employed to detect errors and to determine ways of correcting those errors (Norman, 1983).

User Models

Conceptual and mental models are modeling the system, in contrast to the user model, which describes the *user* of the system. The user model is perhaps the most elusive of the three types of models. Daniels (1986) defines the user model as "the model held by a system of a user" (p. 272). User modeling is based on the notion that any time two individuals interact, they each have a model or knowledge of the other. The assumptions each makes about the other are a key element when attempting to create a system that mimics a human intermediary in the process of interacting with a user. The ability for a system to function in an interactive capacity allows the computer to "get to know" the user, thereby enabling the system to act as a dynamic participant in the information retrieval process.

Psychologists have studied the processes that occur when two people interact and the models that are formed. Newcomb (1961) presents a model of communication suggesting that when two individuals interact, they each have preconceived assumptions of the other. In other words, each knows what she or he thinks, and also has an idea of what the other person thinks. As communication proceeds and new information is presented, each adjusts their attitude of the other, either reinforcing the existing orientation or reassessing the existing attitudes and developing new ones. Over time, if communication is to continue harmoniously, the attitudes of each will become more similar to the other.

Brooks, Daniels, and Belkin (1985) identify the user model as the element that arises out of communication between two people, or in the system's case, a person and a computer. This knowledge improves the interaction between the user and the system, allowing the system to reason and make judgments based on the information provided by the model, so that the system then can modify its actions in accordance with the user's characteristics (Gilbert, 1987). Clowes, Cole, and Arshad (1985) call the user model a "representation of the user in terms of the user's observed and inferred abilities, beliefs, goals, attitudes, and emotions" (p. 36). The user model serves as a means of distinguishing the user's needs and beliefs from those of the intermediary or system. In human-human interaction, the model can be derived from stereotypes, implicit knowledge, extralinguistic cues, nonverbal communication, the user's situation, or a problem description (Brooks et al., 1986).

There are no strict, mutually exclusive categories by which all user models can be defined, nor is there a consensus as to exactly what is to be included in a user model. Characteristics of the user model can vary according to the system, user, and the task being performed. Daniels (1986) compiles a list of characteristics to be included in the user model: user status, user goals, user knowledge of the field, user experience with information retrieval, and user background (employment, residence, academic background, etc.).

GENERAL CATEGORIES OF USER MODELS

Rich (1979) identifies three dimensions helpful to organize the numerous descriptions of user models. User models are composed of a wide array of information about the user and can be implemented in a variety of types of systems. The dimensions present attributes that a user model is most likely to have and are helpful in determining how useful each will be across various types of systems. The dimensions are not exclusive and can overlap with one another. Rich's dimensions are "canonical vs. individual," "explicit vs. implicit," and "long-term vs. short-term." Rich's dimensions and others have been incorporated into the following categories.

Static vs. Dynamic User Models

The first category of user models that Rich refers to as "canonical vs. individual" may also be seen as "static vs. dynamic," distinguishing a static, unchanging model that is embedded in the system from a dynamic model that is different for individual users and changes throughout the session. Finin (1983) refers to the canonical model as a "generic model" since this category assumes a single model for all users.

Static User Models

Static models can be configured as lists of characteristics that form a *stereotype*. Stereotypical models, just as the name suggests, make assumptions about the user based on the type of information received while interacting with the person. Rich (1979) defines stereotypes as "clusters of characteristics" assigned to predetermined groups of users (p. 332). Stereotypes in systems are analogous to scripts, frames, and schema in human cognitive processes (Stillings et al., 1987). They provide information about events that occur frequently and facilitate the predictability of events or behavior. Brooks, Daniels, and Belkin (1985) propose that a standard set of frames can be used to capture the knowledge that human intermediaries have of their users. According to Rich, two types of information are involved in the implementation of stereotypes: facets, which are the user characteristics, and triggers, which can be a word or words that indicate that the user is displaying some of the characteristics of a particular stereotype, and then prompt activation of the appropriate stereotype (p. 333).

Dynamic User Models

The dynamic model changes throughout the session and over a period of time to incorporate new information received from the user, such as increased experience or change in goal. Each particular user model can be saved under a user identification code and retrieved at each subsequent use.

A system that employs dynamic models builds and changes its model based on each individual user's characteristics. Rich (1979) describes the dynamic model as being "built on the fly" (p. 330), since it is created at the time the system is accessed by the user. It is best implemented in situations where users use the system repeatedly. Systems with infrequent users are best equipped with a static model that is designed for the expected user group (Rich, 1983).

Stated vs. Inferred User Models

User models either can be stated by the user or inferred by the system, based on the responses the system receives from the user. Rich (1979) refers to this dichotomy as explicit vs. implicit (p. 331).

Stated User Models

In systems where an explicit model is implemented, the user is presented with questions as to some characteristic, usually knowledge domain or expertise, and from this information the user is assigned a type, such as "expert" or "novice." Gilbert (1987) calls these models "direct" since the user is questioned directly, which may be a more accurate description than "explicit" since information provided by a user is not always fully and clearly stated, as explicit implies. Daniels (1986) views this type of categorization as a user description, and not really a model at all.

Inferred User Models

The implicit model is embedded within the system and is inferred from the actions or responses of the user. The user may be unaware that an inferred model of him/her is at work, since the system does this on its own, and need not ask the user to provide a self description. The stated and inferred models also may work in combination, with a few initial questions for the user to answer, and then the model is built from the user's subsequent actions.

A system utilizing stereotypic models would assign the user to the most fitting stereotype. Each stereotype has information about the most appropriate style of interaction which should be adopted for users of a certain kind, and by monitoring the user's behavior, the system selects a model that most closely resembles that user. The stereotype allows the system rapidly to infer a user model from a small amount of description (Clowes et al., 1985).

The GRUNDY system (Rich, 1979) employs stereotypes to characterize users for the purpose of recommending novels to them. Each stereotype is assigned a group of features which are numerically weighted in order to match the user with a stereotype more accurately. One of the stereotypes utilized in GRUNDY is "sports-person," containing traits such as physical strength and an interest in sports. The "trigger" for "sports-person" is the word "athletic." If the user identifies him/herself as "athletic," the "sports-person" stereotype will be activated (Rich, 1989).

PLEXUS, an expert system employing a user model to provide referral sources for gardening information, initiates its user model by asking the user questions about prior knowledge of the PLEXUS system, experience with gardening, knowledge of gardening information sources, and objectives in using the system. As the interaction proceeds, the system becomes familiar with the individual and adjusts its responses accordingly (Vickery & Brooks, 1987).

Short-Term vs. Long-Term User Models

Another criteria that Rich (1979) incorporates in her categories of user models is the use of short-term vs. long-term information.

Short-Term User Models

Short-term information is concerned with what the user is doing at the time of the session, what goals the user has, or what is being input by the user. An example of short-term interaction is the library patron's use of the online catalog. Patrons will access the system repeatedly with a specific and most likely different goal each time.

Long-Term User Models

Long-term information involves such elements as expertise and knowledge domain, which can be stored and updated in future sessions. This type of model would be applied to users who interact with the system consistently, where over time a model would be tailored to the individual user. An example of this would be an individual with an account that allows remote access to an online catalog. The user would be recognized by his/her account number to facilitate building and maintenance of a user model.

An example of combining the user model dimensions is found in the case of GRUNDY, where a combination of stated, stereotypic, and long-term models is used. The system asks several introductory questions regarding personality traits of the user to begin creating its model, and as the session progresses the model is modified in accordance with the user's response to the selections made by the system. At the end of the session, the model that has been compiled for that specific user is stored, to be retrieved when she/he returns (Rich, 1979).

Problem Description Models

In the information retrieval domain, it is difficult to separate characteristics of the user from characteristics of the user's problem.

Some information retrieval systems attempt to incorporate both user and problem characteristics into one model, while others separate them into independent models.

Belkin, Seeger, and Wersig (1983) approach the development of the problem description as a distinct modeling task, where a model is created to represent user's need or anomalous state of knowledge. In this type of modeling, the system and the user participate in a dialogue to describe the specific problem explicitly, determining what gaps exist in the user's knowledge of the problem, not unlike the interaction between the user and the reference librarian. A "blackboard" type of system is one way of managing multiple models of the information retrieval process. Each model (of the user, the problem, the database, etc.) would post status information to the blackboard, which then determines what actions to perform (Belkin et al., 1987).

SUMMARY OF USER MODEL TYPES

The various types of user models all share the common goal of understanding the user in order to make systems more useful. Each type of model may contain different information and be presented in a different way. The model dimensions and resulting categories are summarized in the following figure.

	STATIC • user categories • unchanging	**DYNAMIC** • user specific • changes over time
STATED • canonical • direct • generic	• single model • multiple stereotypic	• individual • long term
INFERRED • implicit	• multiple • stereotypic • short term • long term	• adaptive • unique • short term • long term
STATED AND INFERRED • combination model	• multiple • stereotypic • long term	• multiple • adaptive • short term • long term

User model categories

The choice of model types to apply is dependent upon a number of factors.

The richness and depth of a user model, for example, will depend on the amount of information the system can gain about the user, whether the information is gathered by questioning the user directly or by inferring it from the interaction. Each has advantages. Accurate information can be gained by direct questioning on topics that the user can express, such as purpose of search, status, and some keywords. It may be less useful in determining system expertise or understanding of the search question, which might, perhaps, be gathered more accurately by a record of actual interaction with the system. Each of these is problematic and depends on how variables such as "purpose of search" and "expertise" are defined. Similarly, user models can be built in more depth if they are long-term models constructed over the course of multiple search sessions than if they are short-term models built only in a single session.

Another factor in choosing the type of model to apply is the relationship between the user and the system. In a public-access retrieval system with infrequent, anonymous users, it may be possible only to build short-term single-session models, as privacy and expediency factors may prevail. In the case of private access systems (e.g., internal corporate systems) with frequent users who must identify themselves to the system, much more elaborate models may be possible.

The scope of the information retrieval system also will be a factor in determining the type of model to apply. Large systems with one or more databases covering heterogeneous subject areas and types of material will require more elaborate modeling capabilities than small databases with homogeneous content.

Similarly, the diversity of the user population will be a factor in determining the type of model required. The designers must determine if they are dealing with a diverse population that falls neatly into several stereotypic categories, in which case stereotypic models may be useful. Conversely, it may be a highly diverse population that is not easily segmented into groups, in which case stereotypes may be difficult to apply and more adaptive models will be required. The simplest case is one with a clearly defined homogeneous user population, less likely in information retrieval applications.

Related factors to consider are whether one model will serve all users satisfactorily, or whether a model should be built for each user who approaches the system. Another issue is frequency of use. Do the users tend to be regular, returning users that would benefit from a model that is saved and tailored to them over time, or is their use brief and infrequent, indicating a model that is short term, or perhaps a static model? If the model is a long-term one, are the types of queries by an individual relatively similar, or do the user's goals change significantly from one interaction to the next?

USER MODELS AND INFORMATION RETRIEVAL

User models clearly have many applications in interactive systems, but they may not be suitable for all tasks and all domains. Most of the environments in which user models have been applied have been more structured than information retrieval, such as computer-assisted instruction or advice to medical patients.

User models are most effective when the task, domain, user population, and user goals are clearly defined. It also is easier to construct user models when the user's goals remain static throughout a session (i.e., results of intermediate stages of interaction do not influence later stages).

Correspondingly, user models are least effective when tasks are poorly defined, when the user population is heterogeneous, and when the user's goals are dynamic (i.e., they change throughout the use of the system).

Characteristics of Information Retrieval

Information retrieval environments vary widely in the degree to which tasks, user populations, and user goals are defined. Information retrieval tasks may be narrow and well defined, as in the case of known-item searching in a very small database of limited scope. They may be broad and poorly defined, as in subject searching of the online catalog of a large collection. The tasks may fall anywhere in between, depending upon a number of conditions, including the size of the database and the clarity of the problem.

The user population for information retrieval is sometimes homogeneous and sometimes has a narrow range of goals. This is most likely to happen with small user groups with known characteristics and goals (e.g., chemical engineers searching a small corporate database on geology for oil exploration). More often they are heterogeneous, as in the range of users and goals on university or public library online catalogs. The population might fall anywhere in between, such as a subset of users (e.g., chemistry faculty) with a subset of goals (e.g., newest items on crystallography).

The stability of user goals varies greatly in information retrieval as well. User goals might be static over the course of user-system interaction, as in the case of finding one item quickly. They might be dynamic, as in the case of subject searching that requires browsing, where the user's own knowledge of his or her information need and of the database changes with feedback from the system.

In general, information retrieval is characterized by a relatively unpredictable range of users and user goals, no matter what the subject domain. Information retrieval systems are much more characterized by the need to respond to unique queries than are other types of interactive systems to which user models have been applied.

Issues in the Application of User Models to Information Retrieval

It is useful at this point to return to Karen Sparck Jones's (1989) explanation of the reasons for building user models: for *effectiveness*—getting closer to the goal of the system, reaching the correct decision; for *efficiency*—getting to the result faster; and for *acceptability*—expressing the result in an appropriate, understandable way. Are these appropriate goals for information retrieval?

Bates (1990) has argued that we should be very cautious about what we automate in the information retrieval process, and not automate functions simply because we understand them. Rather, we should look carefully at what portions of the task are most amenable to automation and which portions are best left under user control. She notes that while the market demands automaticity in technologies such as cars and cameras, a consumer demand remains for stick-shift transmissions and for sophisticated, manually operated cameras (Bates, 1989).

We must ask both whether we understand the information retrieval task sufficiently to construct effective, efficient, and acceptable user models, as Sparck Jones (1989) suggests, and if so, whether it is appropriate to do so, as Bates (1990) asks. Reviews of information-seeking studies suggest that we have only limited models of this complex process (Borgman, 1986b; Fidel & Soergel, 1983; Fenichel, 1980; Penniman, 1975). Information retrieval is a far more complex task domain than most areas in which user models are applied.

User models necessarily reduce the amount of control that users have over the searching process. User models make assumptions about users' goals and intents and make decisions for them. While accurate models indeed are helpful and reduce the burden on the searcher, inaccurate user models may do more harm than good by putting the user in the wrong place in the system or by preventing access to some portions or content of the system.

It is fairly safe to say that user models may be effective for information retrieval in narrow, well-defined task domains with well-defined user populations that do not need to control searching fully.

They are likely to be useful in complex systems that are otherwise difficult to use. One should be cautious, however, in making broad claims for the applicability of user models in information retrieval.

CONCLUSION

User models are a powerful way to add intelligence to an information retrieval system, but information retrieval is a complex task and it is not clear how effective user models will be under what circumstances. Thus user models should be implemented cautiously in well-defined task environments, and experimentation is encouraged. Only then will we know what the benefits and limitations are of user models in information retrieval.

APPENDIX

User Model Example

The following is a hypothetical example of a *static user model* based on *stated* information from the user to assign a *stereotype*. The hypothetical domain is an online catalog in a large academic library, containing 1 to 2 million title records.

The system poses the following questions to the user (answers in caps):

1. What is your academic status: undergraduate, graduate, faculty, guest? UNDERGRADUATE
2. What is the purpose of your search today: class assignment, term paper, work for faculty member, personal interest? TERM PAPER
3. How many times have you used the system before: never, 1-5 times, 6-10 times, more than 10 times? NEVER
4. Are you interested in a general subject area, a very specific subject area, or for a book or journal whose name you know? GENERAL SUBJECT AREA
5. How much searching have you done on this topic already: not looked anywhere, searched journal indexes already, searched other catalogs, collected some books or articles already? NOT LOOKED ANYWHERE
6. Please type in up to 5 keywords that describe your search topic: COMPUTER VIRUSES

The system will assign the user to the following stereotypic model based on the above answers to the questions:

The user is assigned to the novice stereotype, both in use of the system and in the subject domain.

The user is assigned to the subject browsing search stereotype, as he or she needs to develop his or her topic and terminology more fully.

The system will take the following actions based on this user model:

The user will be put into a menu-oriented search mode rather than a command mode that assumes more knowledge of the system.

The user will be put into the subject authority list in the vicinity of COMPUTER VIRUSES to browse for appropriate synonyms or cross references.

The system may also perform a title keyword search on COMPUTER VIRUSES (using variant forms of the phrase) because this is not an authorized LC Subject Heading (LCSH).

The system will limit the user's output to 100 items, assuming that this is a starting point for further research.

The user will be referred to journal literature databases that may be components of the same system or available elsewhere on campus, based on the lack of occurrence of the LCSH and the likely small retrieval.

REFERENCES

Bates, M. J. (1990). Where should the person stop and the information search interface start? *Information Processing and Management, 26*(5), 575-591.

Bates, M. J. (1989). The design of browsing and berrypicking techniques for the online search interface. *Online Review, 13*(5), 407-424.

Belkin, N. J.; Borgman, C. L.; Brooks, H. M.; Bylander, T.; Croft, W. B.; Daniels, P.; Deerwester, S.; Fox, E. A.; Ingwersen, P.; Rada, R.; Jones, K. S.; Thompson, R.; & Walker, D. (1987). Distributed expert-based information systems: An interdisciplinary approach. *Information Processing and Management, 23*(5), 395-409.

Belkin, N. J.; Seeger, T.; & Wersig, G. (1983). Distributed expert problem treatment as a model for information system analysis and design. *Journal of Information Science, 5*(5), 153-167.

Borgman, C. L. (1986a). The user's mental model of an information retrieval system: An experiment on a prototype online catalog. *International Journal of Man-Machine Studies, 24*(1), 47-64.

Borgman, C. L. (1986b). Why are online catalogs hard to use? Lessons learned from information retrieval studies. *Journal of the American Society for Information Science, 37*(6), 387-400.

Brooks, H. M. (1986). Developing and representing problem descriptions. In B. C. Brookes (Ed.), *Intelligent information systems for the information society* (pp. 141-161). North-Holland: Elsevier Science.

Brooks, H. M.; Daniels, P. J.; & Belkin, N. J. (1985). Problem descriptions and user models: Developing an intelligent interface for document retrieval systems. *Advances in intelligent retrieval: Informatics, 8*, 191-214.

Clowes, I.; Cole, I.; & Arshad, F. (1985). User modelling techniques for interactive systems. In P. Johnson & S. Cook (Eds.), *People and computers: Designing the interface* (pp. 35-45). Cambridge, England: Cambridge University Press.

Daniels, P. J. (1986). Cognitive models in information retrieval: An evaluative review. *Journal of Documentation, 42*(4), 272-304.

De Mey, M. (1977). The cognitive viewpoint: Its development and its scope. *Communication and Cognition, 10*(2), 7-23.

Fenichel, C. H. (1980). The process of searching online bibliographic databases: A review of research. *Library Research, 2*(2), 107-127.

Fidel, R., & Soergel, D. (1983). Factors affecting online bibliographic retrieval: A conceptual framework for research. *Journal of the American Society for Information Science, 34*(3), 163-180.

Finin, T. (1983). Providing help and advice in task oriented systems. In A. Bundy (Ed.), *Proceedings of the Eighth International Joint Conference on Artificial Intelligence* (pp. 176-178). Los Altos, CA: William Kaufmann.

Gilbert, G. N. (1987). Cognitive and social models of the user. In H. J. Bullinger & B. Shackel (Eds.), *Human-Computer Interaction—Interact '87* (pp. 165-169). North-Holland: Elsevier Science.

Halasz, F., & Moran, T. (1982). Analogy considered harmful. In *Proceedings of the Conference on Human Factors in Computer Systems*. Washington, DC: Association for Computing Machinery.

Newcomb, T. M. (1961). *The acquaintance process*. New York: Holt, Rinehart & Winston.

Norman, D. A. (1983). Some observations on mental models. In D. Gentner & A. L. Stevens (Eds.), *Mental models* (pp. 7-14). Hillsdale, NJ: Lawrence Erlbaum Associates.

Oddy, R. N. (1977). Information retrieval through man-machine dialogue. *Journal of Documentation, 33*(1), 1-14.

Penniman, W. D. (1975). Rhythms of dialogue in human-computer conversation. (Doctoral dissertation, Ohio State University, 1975). *Dissertation Abstracts International, 36* (3A), 1142.

Rich, E. (1979). User modelling via stereotypes. *Cognitive Science, 3*(4), 329-354.

Rich, E. (1983). Users are individuals: Individualizing user models. *International Journal of Man-Machine Studies, 18*(3), 199-214.

Rich, E. (1989). Stereotypes and user modeling. In A. Kobsa & W. Wahlster (Eds.), *User models in dialog systems* (pp. 35-51). New York: Springer-Verlag.

Sparck Jones, K. (1989). Realism about user modeling. In A. Kobsa & W. Wahlster (Eds.), *User models in dialog systems* (pp. 341-363). New York: Springer-Verlag.

Stillings, N. A.; Feinstein, M. H.; Garfield, J. L.; Fissland, E. L.; Rosenbaum, D. A.; Weisler, S. E.; & Baker-Ward, L. (1987). *Cognitive science: An introduction*. Cambridge, MA: MIT Press.

Vickery, A., & Brooks, H. (1987). Expert systems and their applications in LIS. *Online Review*, *11*(3), 149-165.

AMY WARNER

School of Information and Library Studies
University of Michigan—Ann Arbor

Natural Language Processing: Current Status for Libraries

ABSTRACT

A general introduction to natural language processing is provided, including a definition and an overview of how natural language processing systems work. Representative systems from both the research and applied sectors are presented in order to illustrate the state of the art in the field and the issues which underlie system design and implementation. Actual and potential areas for natural language processing in information retrieval, including retrieval from online catalogs, indexes, and full texts are discussed, with an assessment of short- and long-range agendas and possible limitations.

INTRODUCTION

Applications of artificial intelligence (AI) to library and information science have been investigated since the late 1970s, and have focused for the most part on expert systems as the most relevant area of AI to pursue. The other papers in these proceedings reflect this interest in expert systems research and development, in their coverage of applications areas (including reference, cataloging and indexing, document delivery, and the user interface); theoretical models (user models); and technologies (knowledge representation techniques). This is understandable given that many of the identified tasks exhibit at least some of the characteristics which make them amenable for expert systems development (Brooks, 1987).

At the same time, however, most of the data manipulated in our automated systems are textual. The information systems themselves consist primarily of free-form natural language text, and they are queried using textual representations as well. Given the sheer quantity of text now available to be searched in machine-readable form, it is not surprising that the "management" of that text by both the system designer and the user is becoming an increasingly difficult problem. This seems most apparent with full text of documents, which are particularly difficult to search and browse given current retrieval techniques (Blair & Maron, 1985).

The main assumption of this paper is that one major problem in human interaction with textual databases is linguistic. Thus, whereas it is very important to understand and model the expert system heuristics associated with the retrieval process, it is also crucial to understand, represent, and effectively manipulate the relevant linguistic structures. This is the problem domain for natural language processing within information retrieval, including interactions with both commercial IR systems and online catalogs.

In addition to justifying this basic assumption, this paper also addresses the following themes and issues:

1. The scope of natural language processing and its relationship to artificial intelligence, specifically to expert systems.
2. The basic architecture of natural language processing systems, and some guiding assumptions, both practical and theoretical, of the field.
3. How and where natural language processing can be most usefully applied in information retrieval.
4. The potential and the limits of natural language processing.

NATURAL LANGUAGE PROCESSING (NLP)

The area known as "natural language processing" is one of three fields which are highly related in their merger of certain aspects of linguistics and computer science. The cognate fields which will be defined include computational linguistics, natural language processing, and natural language understanding. Although these terms often mean somewhat different things to different people and are in fact sometimes used interchangeably, an attempt is made here to make valid distinctions among them by discussing their similarities and differences.

Probably the oldest of these fields is *computational linguistics,* which is essentially concerned with the algorithms or formalisms that are used to process language, specifically with their computational power. A major issue involves research into the computational tractability of various linguistic formalisms, a major concern in systems implementation.

Natural language processing is an area of research and application that explores the computer processing of natural language as part of a system that is intended to interact in some way with a user. Input and output may be in the form of single sentences or sentence fragments, or in connected text. Furthermore, language can be entered and retrieved in spoken or written (keyed) form, with this discussion emphasizing the written form.

Natural language understanding is the part of natural language processing which aims at discovering and using knowledge representation techniques from artificial intelligence in language processing systems. These representations are either intended to aid in more flexible, in-depth processing of the linguistic data (an engineering approach) or are intended to serve as psychological models of human language production and comprehension (a cognitive approach) (Hayes, 1978).

Systems which computationally process and manipulate natural language may therefore fall within or outside the AI paradigm depending on whether their algorithms are claimed to "understand" the language being manipulated, in which case they are more accurately referred to as natural language understanding systems. "Understanding" is usually accomplished by using well-known AI data structures such as semantic networks, scripts, and frames.

The term *natural language processing* is used generally to refer to all technologies and systems, both AI and non-AI based, which analyze and manipulate the linguistic data. All natural language processing systems, whether or not they incorporate AI technology, are built to accomplish some linguistic task, such as text understanding, text generation, and natural language interfaces to database management systems or expert systems (Warner, 1987). The term natural language processing is used most frequently in this paper, in which the main point being investigated is the use of the broad range of pure natural language processing techniques in information retrieval systems.

All three areas—computational linguistics, natural language processing, and natural language understanding—have drawn at various times and to varying degrees on work from *linguistic theory*. This is the academic discipline which studies and attempts to formally model the structure of language. The computational power and tractability of these formalisms have been investigated by computational linguists, and developers of natural language systems have sometimes based their processing algorithms on them. However, since there is no language whose structure has been completely formalized by theorists, these models often must be greatly extended or modified by natural language systems developers. There are also systems which are not based on any model from linguistic theory, but whose processing algorithms incorporate new approaches developed from scratch by the AI community.

Another useful distinction to make is between natural language processing and expert systems, since both are often considered components of artificial intelligence and therefore share many things in common. The most important thing they share—that which makes them part of artificial intelligence—is their focus on automating tasks which are believed to require human intelligence. Beyond that, there are some fundamental differences between them which serve to distinguish them as separate enterprises (Mishkoff, 1985):

1. The most overt difference between the two is in their applications. Natural language processing is used to produce natural language interfaces to databases and to process and manipulate the linguistic structures in a text. Expert systems are used to perform the reasoning processes associated with particular technical domains.
2. The domain of natural language processing is human language, whereas the domain of expert systems is some specialized area of human expertise.
3. Language is acquired through a largely unconscious process starting in early childhood, whereas the more specialized knowledge associated with expert systems is acquired later through a conscious learning process. Therefore, the process of discovering the rules to be automated in the two domains is different. The rules of language are indirectly inferred by analyzing linguistic data, whereas the expert system rules are consciously identified by interacting directly with a domain expert in a process known as knowledge engineering.

The preceding discussion implies that these are totally separate areas when in fact they are not. For example, a major effort is underway to endow expert system interfaces with more flexible linguistic capabilities, indicating a merger of the two into one architecturally complete information system (Finin et al., 1986).

Natural Language Processing Systems

Architecture and Issues

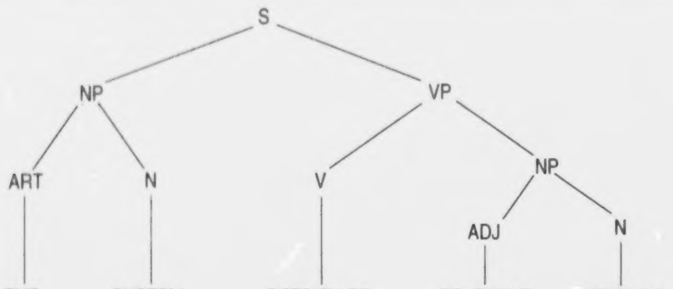
The long-range goal of research and development in natural language processing is to endow a computer with all the necessary rules to process language completely. Underlying this goal are the assumptions that language is systematic and rule-governed; that these rules are discoverable through linguistic analysis; and finally, that the rules, once discovered, are amenable to computational implementation. However, it is clear that the rules of language are both numerous and complex, and the goal of a fully flexible natural language system therefore remains a long-range one. In the short term, parts of the whole problem are being tackled separately in smaller, more manageable systems.

Table 1
Simplified View of a Natural Language System

Natural Language System Component	*Linguistic Level*	*Applied IR Analogy*
Morphological analyzer	Morphological level	Truncation
Lexicon	Lexical level	Stoplist
Parser	Syntactic level	Noun phrases
Semantic analyzer	Semantic level	Thesaurus
Pragmatic analyzer	Pragmatic level	Thesaurus hedges

The first two columns in Table 1 present a highly simplified view of the components of a natural language system and the linguistic levels to which they map; this breakdown is based on the architecture explicated by Winograd (1984). Five levels are described, exemplified by the following description of processing of the simple sentence "The system retrieved relevant articles."

1. *Morphological.* Words (roughly letters bounded by spaces) are decomposed into roots and endings. For example, the term "articles" would be broken into the root "article" and the plural ending "-s." This is accomplished by the morphological analyzer.
2. *Lexical.* Using a dictionary, each root is assigned a set of lexical categories. For example, the stem "article" would be assigned the lexical category NOUN through look-up in a lexicon.
3. *Syntactic.* Using a program module called a *parser,* a structural (i.e., grammatical) description is assigned to the sentence. The program takes the word level input of the lexical component and decides how the individual words go together to form phrases, clauses, and whole sentences. The following analysis of the sentence would result:



4. *Semantic.* The syntactic structure is translated into a form which represents the meaning of the sentence. This involves the determination of the appropriate meaning of each word and then the combination of these into some logical form. This will allow certain inferences to be made about the input. For example, given an appropriate semantic representation, the sentence "The system retrieved relevant documents." would be interpreted as true, since the system would "know" that articles are kinds of documents by interacting with some knowledge representation scheme, such as a semantic network which stores that information.
5. *Pragmatic.* This analyzes the sentence in its context, taking into account a certain body of knowledge about the domain and about the plans and goals of the speaker (user) and hearer (computer) in the conversation. For example, pragmatic information would allow the system to infer that a computer was involved in the retrieval operation, although it is not explicitly stated in the sentence.

The architecture just described should not be considered standard. It is based on the notion that there is so much going on in language that it is necessary to focus on one level of the structure at a time (Crystal, 1987, pp. 82-83). The conception is that a natural language processing system should be modularized—that is, the sentence is processed entirely at one level before being passed on to the next level. This is an intuitively appealing approach, since it allows the designer to work on each smaller component of the system in isolation, and, conversely, to locate and correct errors more easily. However, another more recent approach is described by Allen (1987), in which partial results are passed between modules before analyzing the entire sentence. Furthermore, not all systems contain all the modules delineated above. For example, some combine the syntactic and semantic analysis to produce a semantics-driven parser, as in the system described by Schank and Birnbaum (1984). Finally, systems do not all process to the same depth in terms of linguistic levels, with morphology being the shallowest and pragmatics being the deepest (Weischedel, 1986).

The range of capabilities of current systems is described by Warner (1987), who also provides a summary of issues which have guided research and development in the area. These issues include the following:

1. *Robustness.* Research and development in natural language processing has been oriented toward producing systems with greater depth of analysis and flexibility. Work in this area focuses on processing of ungrammatical or partial input (Carbonell & Hayes, 1984); novel language including metaphor (Carbonell, 1982); and the context of sentences or texts, including the goals and plans of the

participants in a cooperative dialogue (Allen & Litman, 1986; Wahlster & Kobsa, 1986).

2. *Transportability.* Since natural language processing systems can now operate only in limited subject domains, one of the greatest problems is how to best transport techniques used in one subject domain to a new one. This involves not only a system design which makes it transportable, but also a method for customizing it to the new environment (Marsh & Friedman, 1985; Grosz, 1983).
3. *Sublanguage analysis.* At present, some natural language processing systems are being built to process text in small subject domains (e.g., medicine, molecular biology, etc.) characterized by a subset of linguistic patterns—i.e., characteristic constructions. This effectively reduces the number of operations which must be coded and carried out to a manageable size. One long-standing project based on sublanguage analysis is New York University's Linguistic String Project (LSP) (Sager, 1981; Sager et al., 1987). It uses a precise sublanguage description to convert hospital records into a structured format, which can then be used in various applications, including the production of summary reports and question answering.
4. *Ambiguity and synonymy.* A major theme in natural language processing centers around the processing of specific constructions which are known to be either highly ambiguous or synonymous with other constructions. The goal is to endow the system with the capability to generate only one analysis for each linguistic structure (resolve ambiguity), and to generate the *same* analysis for different structures which have the same meaning (eliminate synonymy). This effectively results in a one-to-one correspondence between form and meaning. Constructions in which ambiguity or synonymy need to be handled include compound noun phrases (e.g., FOREIGN STUDENT TEACHING—is this "teaching of foreign students" or "teaching by foreign students"?) (Sparck Jones, 1985; Taylor et al., 1989); coordinate constructions (e.g., OLD MEN AND WOMEN WITH GLASSES—are both men and women old or is it just the men who are old?) (Fong and Berwick, 1985); and paraphrases (JOHN HIT THE BALL/THE BALL WAS HIT BY JOHN—roughly synonymous structures analyzed the same way: JOHN (agent) HIT (verb) BALL (patient) (Harris, 1985, pp. 326-29).

In summary, there is a need within natural language processing to build flexible, cooperative systems based on rules which can be used in other new systems. However, because language is so complex and ambiguous, this can only be done currently in limited domains and only for certain constructions. An important generalization underlies this: There is, in general, a trade-off between the subject breadth of the information contained in the system and the depth of processing which can be performed on that information. Essentially, very deep (i.e., pragmatic) analyses can only be carried out in highly restricted subject domains, while greater subject breadth means that the analysis will be shallower.

Operational Information Retrieval System Parallels

Many of the natural language processing systems just covered could be considered information retrieval systems. Indeed, one of the applications areas within that field is the design of natural language interfaces. However, these serve as front ends to database management systems and expert systems rather than to document retrieval systems. This section surveys the parallels between pure natural language processing and document retrieval from bibliographic databases (online catalogs and indexes) and full-text databases.

It is useful to begin by summarizing the current capabilities and architecture of applied (i.e., nonexperimental) document retrieval systems and interfaces to these systems. This discussion, which is an elaboration of the material in column one of Table 1, is based largely on the overviews presented by Doszkocs (1986, 1987) and also refers to the examples in Table 2.

1. *Morphological level.* IR system capabilities for dealing with morphology include prefix, infix, and suffix truncation operators. This means that the system will "ignore" the affix in its matching process. The examples in Table 2a illustrate this.
2. *Lexical level.* IR systems do not use a lexicon to assign parts of speech to individual stems, as in the natural language processing systems previously described. However, a stopword list is employed to prevent machine indexing of certain function words which are not considered useful for content representation.
3. *Syntactic level.* Structural units above the level of individual words or stems are primarily noun phrases. Noun phrases are found at two places in the retrieval system operation. They are part of the system's inverted indexes, but only if they come from controlled term fields (i.e., descriptors, subject headings, or identifiers). They are also "constructed" at search time through insertion by the user of proximity operators which will allow noun phrase variants to be retrieved. For example, the search term PROGRAMMING (2N) INTERFACE would retrieve documents containing any of the syntactic paraphrases found in Table 2b.

4. *Semantic level.* IR systems allow users to manipulate meaning relationships among terms through certain interactions with the thesaurus. For example, given the equivalence relationship found in Table 2c.1, some retrieval systems will automatically substitute the preferred term in the user's strategy if necessary. Also, given the BT-NT relationship depicted in Table 2c.2, in some systems which contain an online thesaurus users can expand their requests by automatically incorporating terms from the hierarchy.
5. *Pragmatic level.* In some ways, the thesaurus can be said to contain pragmatic information. This is because many of the decisions about the relationships among terms are based on indexing practice. For example, the instruction to index a surgical procedure with an accompanying body part (Table 2d) really pertains to the pragmatic level. Another example of pragmatic information in retrieval systems might be the "hedges," collections of search terms associated with particular topics, which have been found through experience to successfully retrieve relevant documents (Sievert & Boyce, 1983).

TABLE 2
IR DATA FROM LINGUISTIC LEVELS

a. Morphology	Graphic interface/Graphic[al] interface
	Nonlinear operat[or]/Nonlinear operat[ion]
b. Syntax	Programming interface
	Interface for programming
	Interface for computer programming
c. Semantics	1. Syntax/Grammar
	2. Computer interface
	IBM interface
	Macintosh interface
d. Pragmatics	Cataract extraction + Lens, crystalline

Operational information retrieval systems can be discussed in terms of the issues of robustness; domain breadth and processing depth; transportability; and handling of ambiguity and synonymy. Most of them operate in very wide subject domains in which large amounts of textual material are processed and in which a wide variety of linguistic constructions are potentially available to be manipulated. However, processing is quite shallow and usually involves the isolation and storage of individual words (strings of characters bounded by spaces); phrases are only isolated and stored when they have been previously assigned from a controlled vocabulary by an indexer or cataloger. Very few incorporate any semantics, although the thesaurus, if available online, is relevant only to a particular database and contains some information about semantic relationships among terms, such as synonymy and genus-species; however, the thesaurus is frequently not linked to the database and the user must often explicitly select terms from the displays. At the same time, most commercially available "user-friendly" interfaces to online databases have exploited the simple internal structure of the database and similarly employ very shallow linguistic analysis. A typical interface might allow the user to input a string of search terms, which would then be searched by automatically inserting a Boolean or single adjacency operator (Benson & Weinberg, 1988) or by automatically stemming and weighting the individual words in the query (Koll et al., 1984). Only a few, such as Tome-Searcher ("Intelligent Search Software...," 1988), provide for query expansion using a lexicon. However, none provides the range of pragmatic query broadening and narrowing capabilities detailed in the exploratory study of Fidel (1986).

Although commercial systems and interfaces are not very robust (i.e., they do not process very deeply or flexibly), their algorithms are very domain-independent and therefore transportable. This is because they have, except in the cases where vocabularies are linked to the databases, worked by simple surface matching of character strings in queries and documents—such an algorithm makes use of no deep "knowledge" of the linguistic or contextual knowledge of the particular domain.

Ambiguity and synonymy have been major issues in information retrieval, and their resolution is one of the major functions of a controlled vocabulary (Lancaster, 1979, p. 181). Ambiguity and synonymy of natural language search terms in isolation are usually resolved when combined with other terms in the query, and are therefore not considered too problematic in the operational information retrieval environment.

NATURAL LANGUAGE PROCESSING AND EXPERIMENTAL INFORMATION RETRIEVAL

Although much of the current interest in producing more sophisticated IR systems has focused on expert systems development, there has been a historic connection between natural language processing and information retrieval. This was investigated by Masterman, Needham, and Sparck Jones (1959), who stated that:

> An analogy made between library retrieval and mechanical translation is usually made by assimilating library retrieval to mechanical translation. We desire to draw the converse analogy; that is to assimilate mechanical translation to library retrieval. To do this, mechanical translation procedures must be generalized and made interlingual, until they become as general as library retrieval procedures already are. This generalization can be made if the mechanical translation procedure is based on a thesaurus. (p. 917)

This idea reflected the thrust of efforts in machine translation, which was based on look-up of individual words in dictionaries. Although this was very appealing, it was also simplistic and therefore not very successful.

Natural language processing research continued, but instead focused on isolating and manipulating complex linguistic structures for other applications, such as question answering, rather than matching and translating individual words. The work in the area was spawned by the formalism in linguistic theory known as transformational-generative grammar (Chomsky, 1957), which seemed amenable to computational implementation. This sparked a number of attempts within information retrieval to directly automate formalisms from linguistic theory in order to improve system performance in areas such as automatic indexing and automatic abstracting. Surveys exploring the relevance of linguistic theory and information retrieval were conducted by Sparck Jones and Kay (1973, 1977) and Montgomery (1972); actual experimental systems based on linguistic theory were implemented by investigators such as Moyne (1968). Since the 1970s, there has been a trend away from direct implementation of formalisms from linguistic theory in IR systems, and a trend toward the adoption of AI approaches (Sparck Jones & Tait, 1984; Croft & Lewis, 1987) as well as the empirical discovery and development of non-AI algorithms tailored to a given retrieval problem or environment (Salton, 1989; Dillon & Gray, 1983; Schwarz, 1988).

Two important points which are relevant to the role of natural language processing to information retrieval need to be made. First, searches of retrieval systems are usually by some topic and are intended to retrieve a set of documents which match a request; this is in contrast with much of the work in pure natural language processing, where systems are often intended to answer specific questions by retrieving particular facts from a database. Second, it is generally assumed, at least within applied IR, that the subjects of documents and queries can be represented adequately by lists of words and phrases; this contrasts with other natural language processing systems in which the linguistic information in the system often results from the full processing of linguistic units at the sentence level and above (i.e., connected text). These fundamental differences have prompted some (Lancaster, 1972, p. 141; Salton & McGill, 1983, p. 258) to question whether natural language processing is relevant to document retrieval. However, these statements were made in an era characterized primarily by intermediary searching of bibliographic databases.

As useful as most current operational retrieval systems and interfaces are, more recent developments in interfaces for end-users and full-text retrieval have revealed a need for even more powerful retrieval aids. Efforts to produce them have been the major focus of experimental information retrieval, and some investigators are making use of natural language processing techniques in those endeavors. There are four general areas which are of current concern:

- Making the systems "transparent" (Williams, 1986), in which more functions would be delegated to the machine, has become a primary goal of the effort to design powerful interfaces.
- Systems with interface capabilities also could be further enhanced by more robust processing (i.e., phrase as well as keyword indexing) of the underlying free text in titles and abstracts.
- Since retrieval effectiveness does not appear to "scale up" very well to large full-text databases (Blair & Maron, 1985), another major issue has involved the manipulation of these texts into a representation which can be searched more effectively.
- The costs of manual production and application of controlled vocabularies, which have always been high, could be lessened through effective automatic procedures.

Comparison of Experimental and Operational Systems

It is useful to compare experimental and operational information retrieval in terms of the analysis procedures which are employed at the five linguistic levels already described. This results in a view of the state of the art of experimental information retrieval and a vantage point from which to discuss both future directions and limitations.

Morphological and lexical analysis within experimental information retrieval closely parallels the procedures employed within operational systems. Stoplists of terms are used in experimental systems to exclude frequently occurring, primarily function words which are not considered to be indicative of subject content. In contrast with operational systems where users normally have to supply truncation operators, automatic term truncation is virtually standard in experimental systems. Virtually all systems which perform automatic indexing and/or natural language query manipulation make use of a truncation procedure at an early stage in their algorithms. It is important to note that automatic truncation procedures in experimental document retrieval systems do not usually employ a fully developed morphological analyzer, as is the case in most of the natural language processing systems previously described. Instead, they linguistically overgeneralize, using, for example, a list of suffixes to remove the longest matching suffix on the end of a given word; this results in an efficient processing algorithm, although it does produce some processing errors (Salton & McGill, 1983, pp. 72-73).

Syntactic analysis within experimental information retrieval in general focuses on the isolation of noun phrases from free text—titles, abstracts, full texts, and natural language queries. This is achieved through some kind of parsing procedure, although it is not necessary to process any given sentence as fully as in other natural language processing systems. One purpose of a syntactic analysis of this sort is to enable a system to process strings of natural language words input by the user into meaningful phrases; this can be used as a precision device (Metzler & Haas, 1989) or as a method of grouping related phrases for query reformulation (Salton, 1989). Another purpose is to automatically index a document collection using noun phrases instead of the usual keywords (Dillon & Gray, 1983; Schwarz, in press).

Work in incorporating semantics into experimental information retrieval systems has been undertaken for a variety of purposes and in a number of ways. Semantic analysis is useful in retrieval systems since it permits word and phrase manipulation based on criteria other than the matching of surface strings; that is, it is an attempt to manipulate word senses rather than word tokens. Thus, in order for semantic analysis to be accomplished, information regarding the meanings of terms and/or their relationships to each other must be identified, stored, and made available to the system. In one approach, investigators build semantic representations to be used to manipulate queries in interface design; semantic representations built for this purpose are quite varied, and include case frames, fuzzy logic, and semantic word classes (Croft & Lewis, 1987; Biswas et al., 1987; Liddy & Jorgensen, 1989). Systems which attempt to semantically process both the query and the document store are also based on different techniques, ranging from semantic analyzers which manipulate semantic primitives (Sparck Jones & Tait, 1984) to parsing procedures which extensively employ semantic information from available machine-readable dictionaries and thesauri (McCray, 1989). Finally, attempts are being made to use semantic criteria to automatically construct a thesaurus for an information retrieval system from a machine-readable dictionary (Ahlswede et al., 1988).

The pragmatic level has been explored informally by a few investigators. The discourse properties of scientific abstracts have begun to be explored (Liddy, 1988; Liddy et al., 1987), specifically to determine if there is any regular, implicit structure which can be exploited and whether there is any predictable way to determine the anaphoric referents (e.g., the specific entity to which a pronoun refers). The determination of such regular structures would be useful in both searching and automatic indexing procedures. Other projects are experimental systems which deal with natural language input but also consider search strategy formulation and reformulation an expert system task. For example, the

IR-NLI-II system (Brajnik et al., 1987) incorporates mechanisms which allow for the understanding of anaphoric referents and indirect speech as well as the management of a clarification dialogue with the user and the IR system (Croft & Thompson, 1987) creates a model of the user's information need which can be modified based on evaluation of system output and/or a change in the user's goals.

Virtually all of the issues previously identified for pure natural language processing also apply to its application within experimental information retrieval. Automatic linguistic techniques which are generalizable across wide subject domains primarily exist at the lower levels (e.g., morphology and syntax). Systems employing semantic techniques often contain domain-dependent information which usually needs to be hand constructed, making them much smaller and therefore much less extensible to an operational environment. An improvement over this situation is the more recent trend toward exploiting machine-readable dictionary information (Krovetz & Croft, 1989), which enables the system to have at its disposal much more semantic information. Furthermore, systems developers are concerned with many of the difficulties associated with constructions which are either ambiguous or synonymous and therefore difficult to process effectively. These include ambiguous noun phrases (McCray et al., 1988); noun phrases and their paraphrases (Dillon & McDonald, 1983; Salton, 1989; Sparck Jones & Tait, 1984); and conjunctions (Das-Gupta, 1987).

In summary, the following generalizations may be made about natural language processing and document retrieval, including both experimental and operational systems: first, there continues to be a reliance on subject representation techniques by words and/or phrases. Operational and experimental systems differ not in what they represent, then, but in terms of the degree to which they isolate, manipulate, and interrelate these structures. Thus, experimental systems tend to be much "richer" in their processing of syntactic units (noun phrases) and semantic (word and phrase) senses and their relationships. Second, in general, size and domain breadth remain problems within information retrieval because they create an unfortunate trade-off with processing depth. This means that domain independent analysis, generalizable across large numbers of operational systems, remains largely dependent on the matching of surface strings, with surface analyses and matches often not fully accounting for synonymy and ambiguity in lexical items, which often can be resolved only at a deeper, semantic level (e.g., GRAMMAR and SYNTAX from Table 2c.1 cannot be given an equivalence relationship *automatically* through a simple match of any surface elements).

AN AGENDA FOR NLP AND INFORMATION RETRIEVAL

The history of information retrieval has demonstrated that the role for natural language processing within the field is a controversial one. It is still fair to say that natural language processing techniques remain largely a promissory note rather than an accepted and established agenda. Although this can be considered a harsh indictment, it does reflect the larger problem of how to meaningfully relate the procedures and findings of the research domain with what is going on in the applied arena. At the same time, however, it is clear that system performance is still far from perfect (i.e., 100 percent recall and precision) and that other, nonlinguistic techniques have not improved retrieval performance very much (Lewis et al., 1989).

In many ways, one can view the linguistic goal in all information retrieval endeavors as the elimination of linguistic variability—that is, eliminate ambiguity and synonymy of search and index terms and create a one <—> one relationship between term forms and the concepts they represent. In the realm of operational information retrieval, construction of controlled vocabularies and their application can be seen as an attempt to manually reduce this variability. Document representation and searching by uncontrolled natural language have other, complementary benefits over indexing and searching with controlled vocabularies (no difficulties associated with the imposition of an artificial language), but they have some disadvantages as well, particularly that users are given the task of handling the linguistic variability of the underlying text themselves by, for example, supplying all synonyms for a given concept. Lancaster (1979, pp. 284-88) noted the trade-offs between natural language and controlled vocabulary, and recognized the need to control natural language more effectively at search time.

Research in experimental information retrieval can be seen as an attempt to eliminate the same kinds of variability which operational systems have eliminated by using indexing languages (controlled vocabulary) or by relying on the searcher to cope with the variability (natural language). Techniques borrowed from linguistics and natural language processing still seem to offer great promise for discovering and automatically managing and manipulating the variability of the natural language of texts and queries.

A continuing problem is the barrier imposed by the domain breadth/processing depth trade-off. This means that the easiest and most computationally viable linguistic techniques to use within information retrieval remain those from morphology and syntax. It is possible that Lewis, Croft, and Bhandaru (1989) are right—that surface (morphological and syntactic) techniques unequivocally will not result in very great improvements in retrieval performance, and that they should be abandoned in favor of semantics. However, this seems to imply that the relevant linguistic problems in retrieval lie either on the surface (grammatically) or deep (semantically) inside the text. In fact, there do seem to be some phenomena which do lie on the surface, as the examples in Table 2a, b illustrate. Thus, it seems reasonable to suggest that grammar does play a role in some of the linguistic variability which should be accounted for in the retrieval environment; this in fact has been the viewpoint of some within the natural language processing community for quite some time (Marcus, 1984).

Nevertheless, that semantics accounts for much of the language variability in information retrieval has been well documented by Blair and Maron (1985) and Sievert and McKinin (1989) for full text, and by Lesk (1988) for catalogs. Basically, the problem is that, given a particular concept in retrieval, the phrasing of that concept is highly variable; this is not just a syntactic problem, as already pointed out, but can also be a semantic one as the example in Table 2c.1 illustrates. This justifies the techniques which store and manipulate the senses of words and phrases. However, automatic semantic techniques require that elements of meaning and their relationships be accessible from some source. The necessity of deriving this from scratch has made these systems small, and they will probably remain small unless some way is found to produce larger semantic information stores. There are basically two ways of doing this:

1. Extract the semantic information *automatically* from a machine-readable dictionary. There are several machine-readable dictionaries now available, and a current topic of intense investigation within the natural language processing literature involves the automatic extraction and use of information from these tools (Byrd et al., 1987; Boguraev & Briscoe, 1987).
2. Employ large-scale *manual* procedures to explicitly encode certain linguistic features of texts and/or terminology which can then be manipulated automatically. For example, the Unified Medical Language System (UMLS) (Tuttle et al., 1988) is a project sponsored by the National Library of Medicine, which is attempting to build a "meta-thesaurus" of biomedical terminology which can be used to provide a uniform user interface to heterogeneous sources of information.

CONCLUSION

The preceding discussion presented what has legitimately been described as a simplistic view of the retrieval situation. It implies that the user and the system (i.e., the speaker and the hearer) understand concepts and texts in the same ways. Among others, Belkin, Oddy, and Brooks (1982) have advocated the design of retrieval systems which can build models of the user's needs and views of the world and can revise that model based on additional information. Some individuals would call this an expert retrieval system. However, it is clear that this expert system would need large amounts of pragmatic information about what constitutes a cooperative dialogue and about how to diagnose and correct retrieval errors based on notions such as what the user wants from the system and what the user knows about the system.

Thus, an agenda for the development of increasingly sophisticated information retrieval systems which incorporate natural language processing techniques can be proposed. It involves investigation of linguistic phenomena as well as implementation and testing in actual systems of structures from all linguistic levels: morphological, syntactic, semantic, and pragmatic. Furthermore, it seems reasonable, at least in the short term, to implement a strategy in which as much is done on the surface (with morphology and syntax) as possible, leaving semantic and pragmatic analysis for problems which cannot be solved in any other way.

The prospects for making progress in all of these areas now seem better than ever because of the strides which have been made in natural language processing itself, which is now a more mature field; the increasing amount of study within linguistic theory of semantic and pragmatic phenomena (Morgan, 1982) and the structure of texts (Beaugrande & Dressler, 1981); and the favorable climate within information retrieval for a new look at linguistic techniques and what they have to offer (Croft, 1987).

Future Prospects

An open question always remains about how far information retrieval can go with linguistic techniques. The goal of a fully automatic, fully flexible retrieval system may never be realized; however, systems can surely be made *more* flexible, adaptable, and responsive than they are, and we can also learn something about the linguistic structures inherent in texts and queries in the process.

In the end, however, information retrieval, having a large applied component, will judge any product by its utility and not by whether or not it is a system based on expert systems, natural language processing, or any other technology.

> As with any computer product, the value to the user has nothing to do with the underlying techniques used to create the product. The user just wants something to solve a problem, and a product either solves it or doesn't. If it does answer a need, the product must be judged in its effectiveness against other products that solve the same need. (Harris & Davis, 1986, pp. 156-57)

REFERENCES

Ahlswede, T.; Anderson, J.; Evens, M.; Li, S. M.; Neises, J.; Pin-Ngern, S.; & Markowitz, J. (1988). Automatic construction of a phrasal thesaurus for an information retrieval system from a machine readable dictionary. In *User-oriented content-based text and image handling* (RIAO 88 Conference with Presentation of Prototypes and Operational Demonstrations, 21-24 March 1988) (pp. 597-608). Paris: C.I.D.

Allen, J. (1987). *Natural language understanding.* Menlo Park, CA: Benjamin/Cummings Publishing Company.

Allen, J. F., & Litman, D. J. (1986). Plans, goals, and language. *Proceedings of the IEEE, 74*(7), 939-947.

Beaugrande, R. A. de, & Dressler, W. U. (1981). *Introduction to text linguistics.* London: Longman.

Belkin, N. J.; Oddy, R. N.; & Brooks, H. M. (1982). ASK for information retrieval: Part I. Background and theory. *Journal of Documentation, 38*(2), 61-71.

Benson, J. A., & Weinberg, B. H. (Eds.). (1988). *Gateway software and natural language interfaces: Options for online searching.* Ann Arbor, MI: Pierian Press.

Biswas, G.; Bezdek, J. C.; Marques, M.; & Subramanian, V. (1987). Knowledge-assisted document retrieval: I. The natural-language interface. *Journal of the American Society for Information Science, 38*(2), 83-96.

Blair, D. C., & Maron, M. E. (1985). An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of the ACM, 28*(3), 289-299.

Boguraev, B., & Briscoe, T. (1987). Large lexicons for natural language processing: Utilising the grammar coding system of LDOCE. *Computational Linguistics, 13*(3-4), 203-218.

Brajnik, G.; Guida, G.; & Tasso, C. (1987). User modeling in intelligent information retrieval. *Information Processing and Management, 23*(4), 305-320.

Brooks, H. M. (1987). Expert systems and intelligent information retrieval. *Information Processing and Management, 23*(4), 367-382.

Byrd, R. J.; Calzolari, N.; Chodorow, M. S.; Klavans, J. L.; Neff, M. S.; & Rizk, O. A. (1987). Tools and methods for computational lexicology. *Computational Linguistics, 13*(3-4), 219-240.

Carbonell, J. G. (1982). Metaphor: An inescapable phenomenon in natural-language comprehension. In W. G. Lehnert & M. H. Ringle (Eds.), *Strategies for natural language processing* (pp. 415-434). Hillsdale, NJ: Lawrence Erlbaum Associates.

Carbonell, J. G., & Hayes, P. J. (1984). Coping with extragrammaticality. In *Proceedings of the 10th International Conference on Computational Linguistics and the 22nd Annual Meeting of the Association for Computational Linguistics* (2-6 July 1984) (pp. 437-443). Morristown, NJ: Association for Computational Linguistics.

Chomsky, N. (1957). *Syntactic structures.* The Hague: Mouton.

Croft, W. B. (1987). Approaches to intelligent information retrieval. *Information Processing and Management, 23*(4), 249-254.

Croft, W. B., & Lewis, D. D. (1987). An approach to natural language processing for document retrieval. In C. T. Yu & C. J. Van Rijsbergen (Eds.), *SIGIR '87* (Proceedings of the Tenth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval, 3-5 June 1987) (pp. 26-32). New York: Association for Computing Machinery.

Croft, W. B., & Thompson, R. H. (1987). I³R: A new approach to the design of document retrieval systems. *Journal of the American Society for Information Science, 38*(6), 389-404.

Crystal, D. (1987). *The Cambridge encyclopedia of language.* Cambridge, England: Cambridge University Press.

Das-Gupta, P. (1987). Boolean interpretation of conjunctions for document retrieval. *Journal of the American Society for Information Science, 38*(4), 245-254.

Dillon, M., & Gray, A. S. (1983). FASIT: A fully automatic syntactically based indexing system. *Journal of the American Society for Information Science, 34*(2), 99-108.

Dillon, M., & McDonald, L. K. (1983). Fully automatic book indexing. *Journal of Documentation, 39*(3), 135-154.

Doszkocs, T. E. (1986). Natural language processing in information retrieval. *Journal of the American Society for Information Science, 37*(4), 191-196.

Doszkocs, T. E. (1987). Natural language user interfaces in information retrieval. In F. W. Lancaster (Ed.), *What is user friendly?* (Papers presented at the 1986 Clinic on Library Applications of Data Processing, 20-22 April 1986) (pp. 80-95). Urbana-Champaign: University of Illinois, Graduate School of Library and Information Science.

Fidel, R. (1986). Towards expert systems for the selection of search keys. *Journal of the American Society for Information Science, 37*(1), 37-44.

Finin, T. W.; Joshi, A. K.; & Webber, B. L. (1986). Natural language interactions with artificial experts. *Proceedings of the IEEE, 74*(7), 921-938.

Fong, S., & Berwick, R. (1985). New approaches to parsing conjunctions using Prolog. In *Proceedings of the Association for Computational Linguistics 23rd Annual Meeting* (8-12 July 1985) (pp. 118-126). Morristown, NJ: Association for Computational Linguistics.

Grosz, B. J. (1983). TEAM: A transportable natural-language interface system. In *Proceedings of the Conference on Applied Natural Language Processing* (1-3 February 1983) (pp. 39-45). Morristown, NJ: Association for Computational Linguistics.

Harris, L. R., & Davis, D. B. (1986). *Artificial intelligence enters the marketplace.* Toronto, Ontario: Bantam Books.

Harris, M. D. (1985). *Introduction to natural language processing.* Reston, VA: Reston Publishing Company.

Hayes, P. (1978). Trends in artificial intelligence. *International Journal of Man-Machine Studies, 10*(3), 295-299.

Intelligent search system accepts free expression queries. (1988). *Online Review, 12*(1), 6-7.

Koll, M. B.; Noreault, T.; & McGill, M. J. (1984). Enhanced retrieval techniques on a microcomputer. In M. E. Williams & T. H. Hogan (Comps.) *National Online Meeting Proceedings* (10-12 April 1984) (pp. 165-170). Medford, NJ: Learned Information.

Krovetz, R., & Croft, W. B. (1989). Word sense disambiguation using machine-readable dictionaries. In N. J. Belkin & C. J. van Rijsbergen (Eds.), *SIGIR '89* (Proceedings of the Twelfth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval, 25-28 June 1989) (pp. 127-136). New York: Association for Computing Machinery.

Lancaster, F. W. (1972). *Vocabulary control for information retrieval.* Washington, DC: Information Resources Press.

Lancaster, F. W. (1979). Information retrieval systems: Characteristics, testing and evaluation (2d ed.). New York: Wiley.

Lesk, M. (1988). They said true things, but called them by wrong names—Vocabulary problems over time in retrieval. In *Information in text* (Fourth Annual Conference of the UW Centre for the New Oxford English Dictionary, 26-28 October 1988) (pp. 1-10). Waterloo, Ontario: University of Waterloo.

Lewis, D. D.; Croft, W. B.; & Bhandaru, N. (1989). Language-oriented information retrieval. *International Journal of Intelligent Systems, 4*(3), 285-318.

Liddy, E. D. (1988). Structure of information in full-text abstracts. In *User-oriented content-based text and image handling* (RIAO 88 Conference with Presentation of Prototypes and Operational Demonstrations, 21-24 March 1988) (pp. 183-195). Paris: C.I.D.

Liddy, E.; Bonzi, S.; Katzer, J.; & Oddy, E. (1987). A study of discourse anaphora in scientific abstracts. *Journal of the American Society for Information Science, 38*(4), 255-261.

Liddy, E., & Jorgenson, C. L. (1989). Processing natural language for an expert system using a sublanguage approach. In J. Katzer & G. B. Newby (Eds.), *Managing information and technology* (Proceedings of the 52nd ASIS Annual Meeting, 30 October-2 November 1989) (pp. 136-141). Medford, NJ: Learned Information.

Marcus, M. P. (1984). Some inadequate theories of human language processing. In T. G. Bever, J. M. Carroll, & L. A. Miller (Eds.), *Talking minds: The study of language in cognitive science* (pp. 253-278). Cambridge, MA: MIT Press.

Marsh, E., & Friedman, C. (1985). Transporting the Linguistic String Project system from a medical to a Navy domain. *ACM Transactions on Office Information Systems, 3*(2), 121-140.

Masterman, M.; Needham, R. M.; & Sparck Jones, K. (1959). The analogy between mechanical translation and library retrieval. In *Proceedings of the International Conference on Scientific Information* (16-21 November 1958) (Vol. 2, pp. 917-927). Washington, DC: National Academy of Sciences and National Research Council.

McCray, A. T. (1989). Parsing, analyzing and accessing biomedical text. In J. Katzer & G. B. Newby (Eds.), *Managing information and technology* (Proceedings of the 52nd ASIS Annual Meeting, 30 October-2 November 1989) (pp. 192-197). Medford, NJ: Learned Information.

McCray, A. T.; Browne, A. C.; & Moore, D. L. (1988). The semantic structure of neo-classical compounds. In R. A. Greenes (Ed.), *Proceedings of the Twelfth Annual Symposium on Computer Applications in Medical Care* (6-9 November 1988) (pp. 165-168). Washington, DC: IEEE Computer Society Press.

Metzler, D. P., & Haas, S. W. (1989). A syntactic filter for improving information retrieval precision. In J. Katzer & G. B. Newby (Eds.), *Managing information and technology* (Proceedings of the 52nd ASIS Annual Meeting, 30 October-2 November 1989) (pp. 24-27). Medford, NJ: Learned Information.

Mishkoff, H. C. (1985). *Understanding artificial intelligence.* Indianapolis, IN: H. W. Sams & Co.

Montgomery, C. A. (1972). Linguistics and information science. *Journal of the American Society for Information Science, 23*(3), 195-219.

Morgan, J. L. (1982). A linguist's view of artificial intelligence. In D. L. Farwell, S. C. Helmreich, & W. D. Wallace (Eds.), *Perspectives in cognitive science* (pp. 4-10). Champaign, IL: Linguistics Student Organization.

Moyne, J. A. (1968). Information retrieval and natural language. In *Proceedings of the American Society for Information Science* (1-6 October 1968) (pp. 259-263). New York: Greenwood Press.

Sager, N. (1981). *Natural language information processing: A computer grammar of English and its applications.* Reading, MA: Addison-Wesley.

Sager, N.; Friedman, C.; & Lyman, M. S. (1987). *Medical language processing: Computer management of narrative data.* Reading, MA: Addison-Wesley.

Salton, G. (1989). Information retrieval using phrase generation methods. In J. Katzer & G. B. Newby (Eds.), *Managing information and technology* (Proceedings of the 52nd ASIS Annual Meeting, 30 October-2 November 1989) (pp. 3-8). Medford, NJ: Learned Information.

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval.* New York: McGraw-Hill.

Schank, R., & Birnbaum, L. (1984). Memory, meaning and syntax. In T. G. Bever, J. M. Carroll, & L. A. Miller (Eds.), *Talking minds: The study of language in cognitive science* (pp. 209-251). Cambridge, MA: MIT Press.

Schwarz, C. (1988). The TINA project: Text content analysis at the Corporate Research Laboratories at SIEMENS. In *User-oriented content-based text and image handling* (RIAO 88 Conference with Presentation of Prototypes and Operational Demonstrations, 21-24 March 1988) (pp. 183-195). Paris: C.I.D.

Schwarz, C. (in press). Automatic syntactic analysis of free text. *Journal of the American Society for Information Science.*

Sievert, M. E., & Boyce, B. R. (1983). Hedge trimming and the resurrection of the controlled vocabulary in online searching. *Online Review*, 7(6), 489-494.

Sievert, M. E., & McKinin, E. J. (1989). Why full-text misses some relevant documents: An analysis of documents not retrieved by CCML or MEDIS. In J. Katzer & G. B. Newby (Eds.), *Managing information and technology* (Proceedings of the 52nd ASIS Annual Meeting, 30 October-2 November 1989) (pp. 34-39). Medford, NJ: Learned Information.

Sparck Jones, K. (1985). Compound noun interpretation problems. In F. Fallside & W. A. Woods (Eds.), *Computer speech processing* (pp. 363-381). Englewood Cliffs, NJ: Prentice-Hall.

Sparck Jones, K., & Kay, M. (1973). *Linguistics and information science*. New York: Academic Press.

Sparck Jones, K., & Kay, M. (1977). Linguistics and information science: After five years. In D. E. Walker, H. Karlgren, & M. Kay (Eds.), *Natural language in information science: Perspectives and directions for research* (Workshop on Linguistics and Information Science, 3-5 May 1976) (pp. 183-192). Stockholm: Skriptor.

Sparck Jones, K., & Tait, J. I. (1984). Automatic search term variant generation. *Journal of Documentation, 40*(1), 50-66.

Taylor, L.; Grover, C.; & Briscoe, T. (1989). The syntactic regularity of English noun phrases. In *Fourth Conference of the European Chapter of the Association for Computational Linguistics* (Proceedings of the Conference, 10-12 April 1989) (pp. 256-263). Morristown, NJ: Association for Computational Linguistics.

Tuttle, M. S.; Blois, M. S.; Erlbaum, M. S.; Nelson, S. J.; & Sherertz, D. D. (1988). Toward a bio-medical thesaurus: Building the foundation of the UMLS. In R. A. Greenes (Ed.), *Proceedings of the Twelfth Annual Symposium on Computer Applications in Medical Care* (6-9 November 1988) (pp. 191-195). Washington, DC: IEEE Computer Society Press.

Wahlster, W., & Kobsa, A. (1986). Dialogue-based user models. *Proceedings of the IEEE, 74*(7), 948-960.

Warner, A. J. (1987). Natural language processing. In M. E. Williams (Ed.), *Annual review of information science and technology* (Vol. 22, pp. 79-108). Amsterdam: Elsevier.

Weischedel, R. M. (1986). Knowledge representation and natural language processing. *Proceedings of the IEEE, 74*(7), 905-920.

Williams, M. E. (1986). Transparent information systems through gateways, front ends, intermediaries, and interfaces. *Journal of the American Society for Information Science, 37*(4), 204-214.

Winograd, T. (1984). Computer software for working with language. *Scientific American, 251*(3), 130-145.

IRENE L. TRAVIS

Information Services Division
World Bank
Washington, DC

Knowledge Representation in Artificial Intelligence

ABSTRACT

The problems of knowledge representation and use in expert systems and the problems of organizing and searching information in libraries and other bibliographic systems have much in common. There are two basic paradigms for representing knowledge in the knowledge bases of expert systems: rule-based and object-based. Of the two, the rule-based approach has had more publicity, but the object-oriented approach, which will seem more familiar to librarians, is coming to be seen as a necessary complement to rules or even as the more basic system component. One of the principal unsolved problems in knowledge representation is how to provide expert systems and natural language processing systems with more world knowledge, particularly "common sense" knowledge, in order to make them more robust. A major project to build a knowledge base of such basic information is underway at the Microelectronics and Computer Corporation (MCC), a corporation financed by a consortium of American industry to carry out research in advanced computing and computing technology. Since the project represents an attempt to organize a very large and general body of knowledge for use, it can be hypothesized that it will face many of the same problems faced by librarians as they have done the same thing. The project's published goals and achievements at the midpoint of its ten-year life are reviewed from that perspective. Four barriers to such efforts are discussed: (1) the variability of human performance in tasks related to knowledge representation and search; (2) the paradox

of structure; (3) the double-edged nature of the 80/20 rule; and (4) the inertia of an installed base.

INTRODUCTION

One purpose of this paper is to provide a more general overview of knowledge representation (KR) in artificial intelligence (AI) than is provided by the discussion of particular projects in other papers in these proceedings. In addition, while most papers have focused on what AI can contribute to libraries, this paper will turn the topic on its head. It will review an advanced AI research project from the perspective of what librarianship has learned through experience with building and maintaining large complex knowledge bases.

KNOWLEDGE REPRESENTATION

AI began as a set of largely unrelated activities in areas such as game playing, theorem proving, robotics, natural language understanding, expert reasoning, machine vision, and other fields. Some were attempts to model human cognition; others were highly pragmatic. With the high level of activity in AI in the last ten to fifteen years, a body of generally accepted practice has evolved in some areas, particularly in expert systems (Buchanan & Smith, 1989).

Basic Paradigms

When many people think of AI, they immediately think of rules. This association stems from the fact that some of the most publicized AI programs and early expert systems, such as MYCIN (Shortliffe, 1976), used rules as their *knowledge bases* (KB). A KB is the body of subject knowledge that supports the performance of a "knowledge-based system," such as an expert or natural language processing system.

MYCIN was also the model for at least the initial versions of many commercial expert system shells. *Shells* are programs that contain the facilities for constructing an expert system, but which do not contain any subject knowledge when sold. They are a fairly exact equivalent of a database management system (DBMS) except that they support expert systems instead of databases. Because of the widespread use of shells and the relative ease of dealing with knowledge in the form of English-like rules, rule-based systems have enjoyed considerable popularity.

Rules easily translate information about how to do things. That is, they are "procedurally oriented," while information about objects in their domain of interest is not pulled together, but scattered and embedded throughout the rules. Not surprisingly, the complementary view of the world represented in many KR schemes is object-centered or "object-oriented." Objects may be things, but they may also be actions or events. Objects may be created for classes or for individuals. They can contain procedural information, but such information is attached to the objects. These two methods for organizing knowledge, procedurally based and object-based, are complementary, rather like alphabetical and classified arrangements in indexing languages. Although there are some exceptions, the same information can generally be expressed under either paradigm, but certain kinds of information are easier to create, control, modify, search, and use in one format than in the other. Many shells that initially had only rule-based capabilities now, therefore, also include facilities for creating and using objects. For an excellent current overview of AI written for the nontechnical reader, the author highly recommends the new textbook, *Computers and Thought* (Sharples et al., 1989).

The Role of Relations

Regardless of which paradigm is adopted, relational analysis is an essential component. When one looks at a rule stated in natural language, for instance,

> Some restaurants take reservations.

the importance of relational analysis may not be obvious, but rules must, in fact, be translated into a formal language, whether into lists for LISP programs, the structure of some other programming language, or the predicate calculus. As soon as such rules are placed in such formal structures, relational analysis is necessary.

Relations may link an attribute to an individual (Crowded, Chez Pierre); an individual to a class (Restaurant, Chez Pierre); or describe the relationship between two or more individuals or classes (EatsAt, John, Chez Pierre). Relations are always stated in a particular order, since they are not usually commutative.

Suffice it to say that it is this emphasis on relational analysis that provides the first strong indication that KR for AI and the traditional pursuits of librarians constructing classification schemes and thesauri are very closely linked. Relational analysis is essential to all index language construction, involving, as it does, the identification and specification of links among concepts. Although it may be carried out at varying levels of detail and complexity, from minimal synonym identification to the role analysis in the PRECIS indexing language, relational analysis is at the heart of all exercises in vocabulary control.

Indeed, indexing languages are representations of knowledge. The range of kinds of knowledge represented is limited to concepts, and, moreover, the relations represented between concepts are relatively undifferentiated, usually being confined to synonymy and hierarchical relationships, with all others being lumped together as "related terms." The representation only of concepts contrasts to the representation of procedures or other dynamic or sensory information in AI.

However, within this restricted scope, librarians have had considerable achievements. They have developed and maintained some very large and complex knowledge bases, and they have in some cases maintained them for over 100 years. Even many thesauri that are considered "modern," in that they were developed for coordinate indexing systems instead of for card catalogs or for shelving books, have been in use for twenty-five or thirty years. No expert systems have that kind of history. Indeed, few have been maintained at all.

Specific Methods

Some specific methods for KR will now be considered: first, rules and their formalization into first-order predicate calculus; then, two object-oriented methods—semantic networks and frames.

Procedurally Oriented Knowledge Structures

The discussion of different KR methods will focus on an example of procedurally oriented information that was initially stated as two rules that might be part of a restaurant selection system (Figure 1). These rules are then translated to two procedurally oriented approaches. It is very natural to express procedural knowledge in such rules. If someone were explaining how restaurant reservations work, he or she would indeed explain it this way. Of course, the rules cannot actually be used as they appear in Figure 1, part (a).

Figure 1, part (b) shows a translation of the rules into first-order predicate calculus, which is familiar to many people from their general education, to show how they can be converted into a formal structure. The use of predicate calculus allows formal theorem-proving techniques to be used in ascertaining the truth of a proposition. Notice that the natural language relations appear as predicates with variables, e.g., Restaurant (x); Patron (y); EatsAt (x,y). Formalisms like predicate calculus delineate the relations inherent in the natural language text and the logical operations among them (AND, OR, NOT, entailment). Moreover, they allow the introduction of quantification, utilizing the universal quantifier "For every" ($\forall$) or the existence quantifier "There exists" ($\exists$).

(a) Knowledge Represented in Rules

Rule 1: If a restaurant accepts reservations and a patron makes reservations at the restaurant, then the patron eats at the restaurant.

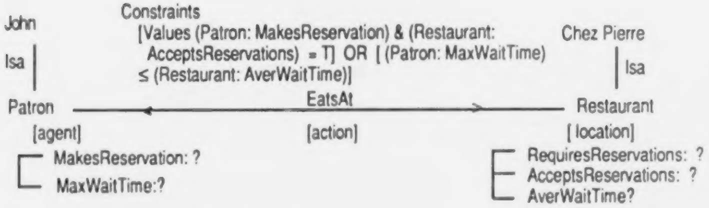
Rule 2: If a restaurant does not require reservations and the maximum waiting time of the patron ≤ the average waiting time of the restaurant, then the patron eats at the restaurant.

(b) Knowledge Represented in Symbolic Logic

P1: $(\exists x\ (\text{Restaurant}(x)\ \&\ \text{AcceptsReservations}(x)))\ \&\ (\forall x\ (\exists y\ (\text{Patron}(y)\ \&\ \text{MakesReservation}(y,x)\ ))\rightarrow \forall x\ \forall y\ (\text{EatsAt}(y,x))$

P2: $(\exists x\ (\text{Restaurant}(x)\ \&\ \perp \text{RequiresReservations}(x)))\ \&\ (\forall x\ (\exists y\ (\text{Patron}(y)\ \&\ (\text{MaxWaitTime}(y) \leq \text{AverWaitTime}(x))\rightarrow\forall x\ \forall y\ (\text{EatsAt}(y,x))).$

(c) Knowledge Represented in a Semantic Network


John
Isa
Patron
[agent]
MakesReservation: ?
MaxWaitTime:?
Constraints
[Values (Patron: MakesReservation) & (Restaurant: AcceptsReservations) = T] OR [(Patron: MaxWaitTime) ≤ (Restaurant: AverWaitTime)]
EatsAt
[action]
Chez Pierre
Isa
Restaurant
[location]
RequiresReservations: ?
AcceptsReservations: ?
AverWaitTime?

(d) Knowledge Represented in Frames

Restaurant
FrameNo.: 0 001
AcceptsReservations: T
RequiresReservations: F
AverWaitTime: 30 min
Instances: Chez Pierre, ...

Chez Pierre
FrameNo.: 0002
AcceptsReservations: F
RequiresReservations: F
AverWaitTime: 20 min.
Isa: Restaurant

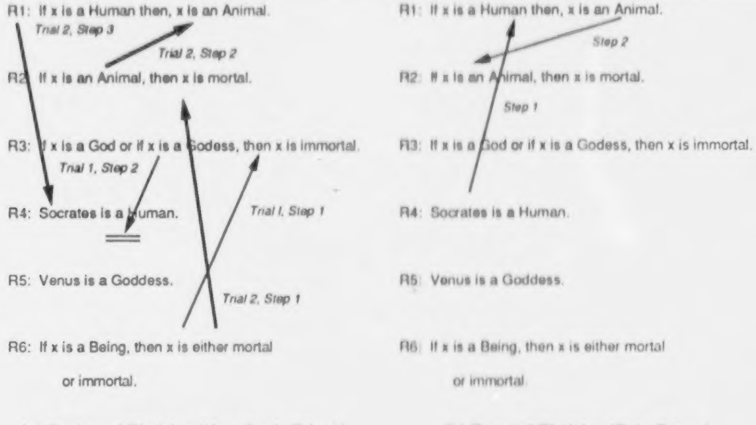
EatsAt (47)
FrameNo.: 0101
Agent: John
Location: Chez Pierre
Constraint (1) T
Isa: EatsAt

EatsAt
FrameNo.: 0005
Agent: Patron
Location: <Restaurant, Cafeteria, SnackBar>
Constraint (1):(Patron: MakesReservation= Restaurant:AcceptsReservations=T) OR Patron: MaxWaitTime ≤ Restaurant: AverWaitTime
Isa: Action
Instances: EatsAt (1); EatsAt(2) . . .

Patron
FrameNo.: 0003
MakesReservation: T
MaxWaitTime: 20
Instances: John, ...

John
FrameNo.: 0004
MakesReservation: [T]
MaxWaitTime: [20]

Figure 1. Alternative forms of part of a knowledge base to decide whether John will eat at Chez Pierre


R1: If x is a Human then, x is an Animal.
Trial 2, Step 3
Trial 2, Step 2
R2: If x is an Animal, then x is mortal.
R3: If x is a God or if x is a Godess, then x is immortal.
Trial 1, Step 2
R4: Socrates is a Human.
Trial 1, Step 1
R5: Venus is a Goddess.
Trial 2, Step 1
R6: If x is a Being, then x is either mortal or immortal.
R1: If x is a Human then, x is an Animal.
Step 2
R2: If x is an Animal, then x is mortal.
Step 1
R3: If x is a God or if x is a Godess, then x is immortal.
R4: Socrates is a Human.
R5: Venus is a Goddess.
R6: If x is a Being, then x is either mortal or immortal.

(a) Backward Chaining (Hypothesis Driven) with Two Hypotheses Tried

(b) Forward Chaining (Data Driven)

Figure 2. Methods of reasoning for concluding that Socrates is mortal, given that Socrates is a human and some type of being

Quantifiers are important in eliminating the ambiguity in many natural language statements. For instance, consider the proposition "Every patron eats at a restaurant." When the proposition is taken out of context, it is not clear whether (1) for every patron there exists a restaurant such that the patron eats there, or (2) there exists a (single) restaurant such that every patron eats at it. Quantifiers can express such differences precisely. Higher order logics can also be used to reason about cause and effect or possibilities.

Object-Oriented Knowledge Structures

Turning to the object-oriented approach, semantic networks, as pioneered by Quillian (1969), are structures that link concepts in a graph and may label the links as to type. Those familiar with the PRECIS indexing language will immediately feel at home with the diagram in Figure 1, part (c), where the horizontal links assign roles to concepts and the vertical ones provide hierarchical information. Objects can also have attributes, such as MakeReservation. What must be added is the procedural knowledge that was so easily stated in the rule-based approach. It is represented here as constraints on the kind of individuals that can serve as specific instances of patron and restaurant and is attached to the EatsAt object.

The same information may be translated quite exactly into frames (Figure 1, part [d]). A frame is created corresponding to each object from the semantic network, including the action EatsAt. Each object stores the attributes and constraints that were shown associated with the object. However, each frame also contains the hierarchical links between it and the broader classes of which it is a member, here shown as "Isa," literally, "is a" slots. Slots in the children of a frame (for instance, the "John" frame is a child of the "Patron" frame) can then be "inherited" from the parent frame, as can default values for these slots. For greater precision, each slot in a child frame could contain a specific "Inherits from" instruction, in case the frame has multiple parents and different slots (attributes) inherit slot fillers (values) from different broader entities. Articles by Susanne M. Humphrey in these proceedings and elsewhere (Humphrey & Kapoor, 1988) illustrate this practice.

As in indexing languages, a list of the narrower terms or specific instances can also be added to a frame, shown in Figure 1, part (d) in the Instances slots. However, these values could also be computed by the system.

The inheritance of slots and slot fillers is very important. Notice that the system has been set up with the information that patrons will make reservations and that the maximum time they are likely to wait is twenty minutes. If the system has no information specifically about John and has no source to ask, then it can continue its processes by having the object for John use the information specified for Patrons as a class as a default. Frame 0004 shows the values inherited from Frame 0003 in brackets. If the system obtains specific information about John, that information overrides the inherited values. This approach is directly analogous to adding narrower terms automatically to a search statement in bibliographic systems. It also provides for considerable space saving, since the default values do not have to be repeated for each object that shares them.

Inheritance cannot be carried out in the rule-based system, even though hierarchical relationships can certainly be expressed, as will be illustrated presently. This limitation clearly makes for considerable inefficiencies in storing information about objects. Notice, on the other hand, that the quantification that was expressed in the predicate calculus example has been lost in the object-oriented approach and, thus, a certain potential for precision in expressing propositions. Nonetheless, the greater part of the information could be expressed in both systems, so both approaches are possible. They can be used exclusively or in conjunction with each other.

One should not conclude this methods section without also briefly mentioning more complex structures than rules or objects. While rules and objects are powerful ways to present knowledge, they do not help very much in conveying the complicated sequences of events associated with many common activities, such as eating in a restaurant or going to work. Many AI systems, therefore, also incorporate scripts or scenarios, which, though they will have rules and objects as components, provide information about the usual order and type of actions to be expected when some commonly occurring event takes place. The entire role of making reservations in eating at a restaurant would be dealt with in a restaurant script along with many other things, such as the role of waiters and menus and money. For physical systems, qualitative models are also a possible way to represent more complex structures. They will be discussed in more detail presently.

Reasoning vs. Knowledge

As stated earlier, the KB is not the whole of an expert system. In fact, modern practice carefully distinguishes it from other components. In expert systems, the most important of these are usually the interface, which will not be discussed here, and the so-called "inference engine" or reasoning mechanism, which will be briefly reviewed to distinguish it from the KB itself.

Inference processes will be illustrated using a somewhat simpler example than restaurant reservations, but one involving more rules. As shown in Figure 2, suppose one is going to use rules stored in a KB to investigate the perpetually intriguing question of whether Socrates is mortal. How can it be done? While there are various special reasoning mechanisms that are invented to handle particular problems, there are two general and widely used methods for reasoning with a KB: (1) backward chaining, also called "hypothesis-driven" reasoning; and (2) forward chaining, also called "data-driven" reasoning.

An instance of backward chaining is shown in Figure 2, part (a). Knowing through rule R6 that Beings can only be mortal or immortal, the system would proceed to adopt each of these hypotheses in turn and see if either could be proven. In this instance, the system begins with the hypothesis that Socrates is immortal (Trial 1, Figure 2, part [a]). Then, it looks for rules having something about immortals in their consequent (the clause following the "then"). In other words, it looks for rules that specify something about the conditions for being an immortal. Finding R3, it then adopts its antecedents (specified in the "if" clause) and searches to see if they are the consequent of other rules. Since the hypothesis that Socrates is immortal cannot, in fact, be proven, the system then adopts the hypothesis that Socrates is mortal as a second trial and begins looking for rules having something about mortals in their consequent. As shown by the steps in Trial 2, this process eventually leads to R4 and the fact that Socrates is a human, which satisfies a condition for being an animal (R1) and, therefore, a mortal (R2), and thus proves the hypothesis.

Alternatively, in forward chaining one starts with a known fact—in this case, that Socrates is human—and then tries to conclude that Socrates is mortal. If an antecedent matches the set of known facts, a rule can be activated or "fired." Knowing that Socrates is human would lead to R1—that humans are animals—and from there to R2—that animals are mortals, also answering our question.

In the example given, it appears that forward chaining is much more efficient than backward chaining, which is not necessarily the case. There could be many other rules in the KB about humans or animals that might have been abortively investigated before the proper sequence of rules was found. Either forward or backward chaining may be preferable, depending on the kind of problem being solved.

It is also worth noting that in this very simple example, almost all the relations are hierarchical ones. It therefore also demonstrates very clearly the potential advantages of an object-oriented approach. If the information in these particular rules had been recorded as a set of hierarchically linked objects (Figure 3), the object for Socrates would have inherited the slot "Life-expectancy" and the slot-filler "mortal" from the object for animals, and the question could have been answered by a simple lookup, instead of the search sequence described. Inconsistencies and gaps in the KB would also have been easier to detect and correct.

The example is oversimplified in other ways. All the facts in the KB are presumed to be true and none of them is inconsistent or contradictory. What the system is doing is attempting to establish a chain of facts that would allow it to *deduce* that Socrates is mortal. But much of human reasoning is not so certain or so strictly logical. One might notice that Socrates was aging and suspect that he was not immortal (abduction) or think that, since all other humans one had observed were mortal, Socrates was probably mortal too (induction). Abduction and induction are perfectly acceptable, practical methods of dealing with the world, even if they are not guaranteed to produce true results.

One can more clearly see this kind of knowledge and its use in the restaurant example above. There, the selection of the matching rule—that a patron will eat at a restaurant if his or her average wait time is less than the maximum wait time of the restaurant—is clearly only a heuristic, a rule of thumb, not a physical law. One of the strengths of knowledge-based systems is that they also attempt to resolve problems in the face of uncertain or insufficient information. One method for augmenting a KB to support such reasoning with uncertainty is to have weights associated with the facts or rules in the KB, usually referred

Animal
  Frameno: 0004
  Isa: Being
  Life-expectancy: Mortal
  Instances: Siamese Cat, Human

Human
  Frameno: 0005
  Isa: Animal
  Instances: Socrates
  Life-expectancy: *<Inherited from Frameno: 0004>*

Socrates
  Frameno: 0006
  Isa: Human
  Life-expectancy: *<Inherited from Frameno: 0004>*

Being
  Frameno: 0001
  Instances: Animal, God, Godesses

Goddess
  Frameno: 0002
  Isa: Being
  Life-expectancy: Immortal
  Instance: Venus, Juno,

Venus
  Frameno: 0003
  Isa: Goddess
  Life-expectancy: *<Inherited from Frameno: 0002>*

Figure 3. Rules from Figure 2 represented as objects

to as "certainty factors." Such weights must then have rules governing their combination.

Many different rules can be used for combining certainties in this way, such as taking the minimum of two certainty factors or multiplying them like probabilities, if they are on a scale (0 to 1). These rules, unlike the weights, would be part of the inference engine, not of the KB. The boundaries between the KB and the inference engine are, thus, carefully maintained.

Problem Areas

There are, of course, a great many unsolved problems in KR. Three particular problems will be mentioned here: KB quality, general or common sense knowledge, and machine learning.

KB Quality

A number of topics may be included under this heading. One is, of course, the accuracy and validity of the knowledge in a KB and how one determines either of these factors: in short, how one debugs the knowledge in a KB. There are several different problems in this category, the first being determining if the system's answers correspond to the expert's in all cases, and the second being the question of whether the expert is right. Even the first, given the nonalgorithmic nature of knowledge-based systems, their ability to modify their own information, and the difficulty in controlling the problems they will be set, presents major barriers. Needless to say, if the area is one, such as bibliographic searching, in which there is not a single right answer against which to measure the results, the judgment of quality is difficult.

The more subjective the judgments involved, the more challenging it is to apply expert system technology: thus, the seemingly redundant admonition that in order to have an expert system, you must have an expert—or put another way, in order to have an expert system, someone must know how to solve the problem.

General Knowledge and Common Sense

One of the principles now recognized by AI knowledge engineers is that it is often useful to distinguish different levels of knowledge: problem specific, domain specific, or general. In an expert system, a considerable degree of performance can often be achieved by including only problem-specific knowledge, if the system is built to solve a very narrow problem and its problem input can be carefully controlled.

However, in many applications it is difficult to limit a knowledge-based system, even an expert system, to problems that require only problem-specific or even only domain knowledge. Without broader knowledge, the range of capabilities of a knowledge-based system is very limited, and the danger that it will be used beyond this range is always present. One pressing problem in AI is how to endow systems with a capability equivalent to the human one (admittedly fallible) of knowing what one does not know. Such a capability, as well as any general extension of system functions, requires the addition of domain or general knowledge.

Furthermore, there are many kinds of AI systems where it is difficult to maintain stringent limits on the problems that will confront the system, for example, in one that supports planning for military operations. To take another example, it is impossible to develop a system to understand fully even quite restricted sorts of text input from uncontrolled sources without having to equip it with much more than problem-specific knowledge. One can, of course, build a system that only attempts to extract and utilize some previously defined information of interest, such as a system that "reads" newspaper stories about terrorist incidents and extracts the basic facts (Lebowitz, 1980). But such a program will ignore or fail to interpret correctly anything that falls outside the scope of its particular filters or its world view.

While such limits are also present in humans, the difference is qualitative as well as quantitative. Such systems have not only a much more limited world view, but also no general principles to bring to bear on new problems. Moreover, they generally cannot learn from their experience or mistakes, since machine learning is still a very young science. Many potentially useful applications for libraries immediately confront the daunting breadth of knowledge required to duplicate the expert behavior of librarians, who draw on a wide base of general and subject-specialized knowledge of the world, in addition to their own professional expertise and techniques for information description and search.

General knowledge is a particularly severe problem in AI, overwhelming both in the quantity of it that even the most ignorant human possesses, and its variety and complexity. Its quantity suggests cooperative efforts to build general KBs that could be used by multiple systems, such as the Cyc Project to be examined presently. Other cooperative efforts related to AI have also been proposed, for instance, the project advocated by Walker (1989) and others to build cooperatively a large corpus of text for natural language processing and promote the sharing and reuse of lexical resources. Of course, any such effort depends on a significant degree of agreement on the contents, structure, and future use of such a tool, which is not easily obtained at this stage of knowledge-based system development. One might compare the effort to the MARC format, which was a necessary prerequisite for a cooperative effort to convert library catalogs to machine-readable form, but which is an expensive data format because of its comprehensiveness and generality.

The attempt to deal with the complexity of general knowledge has opened up many fascinating problems, some of which have been treated for centuries in philosophy—particularly in logic, ontology, and epistemology, and others of which are relatively new as formal, defined, immediate problems. In the first category are such things as the representation of time, causality, possibility, probability, and belief, and the identification and naming of classes of matter, things, qualities, and actions in the world. In the latter class is the AI problem known as "common sense reasoning," including such subfields as qualitative modeling.

The problem addressed by qualitative modeling is, at base, one of the appropriate levels of detail of knowledge for reasoning for a particular application. Humans, for instance, know a lot about when a particular surface will be slippery relative to what they are wearing on their feet. Although one may sometimes be ambushed in an unfamiliar situation, such as nylon socks on carpeted stairs, humans are fairly good at predicting trouble. One knows about such things as hard, shiny surfaces, liquids on surfaces, ice and the relative traction afforded by bare feet, socks, new shoes, leather soles, rubber soles, cleats, or crampons, but how should this knowledge be represented in a KB? Should the KB have hundreds of specific rules about ice, wet floors, waxed tiles, etc., or, at the other extreme, should it have a set of equations for computing the exact degree of friction between the soles of running shoes and the ice on a sidewalk?

Neither approach seems reasonable. One does not allow for any generality of observation and the other requires information that is probably not available and certainly does not represent the way one ordinarily reasons about the problem. Even if the necessary information were available, there may be no analytically tractable or computationally feasible solution.

Qualitative models seek to find a middle ground by reasoning with qualitative information rather than specific quantities. Substances, for instance, are hotter or colder, more or less slippery, smoother or rougher than other substances. Water on a surface increases its slipperiness, but the exact quantities are not measured or estimated.

Although this example involves common sense reasoning, the same kinds of issues apply in representing expert knowledge. Problem- or domain-specific rules are all right as far as they go, but robustness requires that the system also have some general models to fall back on. Even in scientific areas such as medicine or applied geology, experts use a mental model with appropriate levels of detail and appropriate simplifications. A recent stimulating article by W. J. Clancey (1989) has even suggested that all KBs for expert systems could be thought of as qualitative models, and that such a view would allow their power to be compared and assessed against a common scale.

The problems of general or common sense knowledge are particularly crystalized in the continuing debate about the relative roles of grammar, semantics, pragmatics, and world knowledge in the understanding of language. The problem, stated succinctly, is that humans add a lot of background information in interpreting text. Consider a small vignette like the following:

> When Joe's alarm clock went off, he looked out the window.
> Everything was covered with ice. He went back to bed.

In interpreting even this very simple story, one adds to the facts stated that Joe was probably going somewhere because he had his alarm clock set and because the first thing he did was to look outside. One also knows that when he saw the ice, he knew he would have a traction problem; thus, it was dangerous to go out. Also, whatever Joe was going to do was not worth risking his life for. (For a recent review of the state of the art of natural language processing, see Allen, 1989.) As this example clearly demonstrates, the ability to parse sentences and assign dictionary meanings to words is not by any means sufficient to allow the interpretation and full understanding of even short pieces of text. It must fit into some situation, event representation, world model, or other knowledge structure that gives it context and allows its full meaning to be extracted. The scripts described above are one approach to handling stereotyped situations, but many very serious problems remain.

Machine Learning

Finally, it is impossible to discuss problems in the future of knowledge-based systems without mentioning machine learning. Most people working in the field think that it will be impractical to have AI on a large scale unless machines can learn like humans—from experience, from teachers, or from reading. If one wants a real challenge, think of a machine that can learn from watching television!

There are a number of different kinds of learning, some of which involve generalization or abstraction. To take only one case, consider what is involved in having a machine learn by example. Returning to Joe and his running shoes, one would like the machine to be able to generalize about the outcome of walking on slippery surfaces after it had been presented with a number of instances of accidents occurring to people walking on ice, newly waxed floors, etc. However, this exercise requires that the machine extract the essential property of all the surfaces (let us call it slickness-when-walked-on-with-normal-footwear) and, moreover, understand the cause-and-effect relationship between slickness and the accidents.

This observation, of course, brings us back to our qualitative model. Is it possible for a machine to construct this model for itself, and, if so, on what basis? After all, humans have the actual physical experience of having our feet go out from under us from which to learn. The machine does not have the same kind of sensory input. It must also be able to recognize the common characteristic or set of characteristics in the examples in order to be able to generalize from them. Such recognition is a very difficult task for a machine, and one that is, essentially, classification. Indeed, librarians familiar with automatic classification and numerical taxonomy will be interested to know that these same techniques which were introduced in that field twenty-five years ago are now being tried for machine learning.

A MAJOR RESEARCH PROJECT FROM A LIBRARIAN'S PERSPECTIVE: THE CYC PROJECT

The above section has supplied some of the basic information needed to understand an AI application and to put an advanced research project in KR, such as the one to be discussed in the rest of this paper, in some context. It is now time, therefore, to move to the specific example promised.

The reasons for selecting this example, the Cyc Project at the Microelectronics and Computer Corporation, will quickly be obvious. The Cyc Project is the most ambitious attempt now underway anywhere in the world to build a very large and very general KB. In fact, the researchers at MCC propose to build a core KB of about 10 million entries which would then be cooperatively expanded to an unknown size. This KB would contain the "consensus reality" that one needs in order to understand everything in a newspaper (including the ads, advice columns, etc.) and everything in a desktop encyclopedia. In other words, it tackles head-on the problem of world and common-sense knowledge which has been previously described as such a barrier to AI. Since many librarians specialize in organizing large collections of very broad coverage, these facts should immediately capture their interest.

One must admit, nonetheless, that it is also a "convenience sample" of one for the author, since the researchers published a substantial book (Lenat & Guha, 1989) on their experience and progress during the first five years of the project. All comments in this talk are based on this source.

In the technical terms discussed above, the Cyc KB utilizes both frames and predicate calculus for KR. The predicate calculus is used to express constraints, such as "Twins are not likely to have the same first name." It is the more powerful of the two representation forms and includes variants on the universal and existence quantifiers discussed above, except that, in this case, the domain over which the quantifiers operate is always specified. In other words, there are no expressions of the type "For every," but only of the type "ForAll<members of a specified set>."

As was shown in Figure 2, the knowledge that is represented in each slot of a frame can also be considered a predicate. Despite this redundancy, Cyc retains the frame language because it provides a very efficient way to deal with one- and two-place predicates, which constitute the bulk of the information to be stored.

In order to develop the KB, it has been necessary to work out an extensive ontology, that is, to make decisions about what kinds of beings are to be represented in the universe of the system and what relations among them will be recorded. Since this ontology includes about two dozen different classes of things, such as SomethingOccurring, TangibleObject, IntangibleStuff, which have a complicated set of interrelationships, the reader must consult Lenat and Guha for a description. Probably the bulk of the intellectual effort to date has gone into this analysis, which reminds one of similar analyses carried out in the pursuit of universal faceted classification systems in libraries (see, for instance, Dahlberg, 1988). Closely allied with the ontology are the specialized inference mechanisms in the Cyc constraint language, CycL, of which there are more than a dozen.

The system does not use numerical certainty factors for reasoning with uncertainty because, after an initial experiment, the researchers concluded that they tended to lead to too many false inferences. This problem arises because of the subjectivity involved in assigning highly differentiated weights with no objective standards. (Does anyone recall similar criticisms of manually weighted indexing?) Instead, it uses five truth values: T = default true; 100 = monotonically true; — = default false; 0 = monotonically false; and ~ = unknown. "Monotonically true" is assigned to statements whose conclusions must be true if their antecedents are true, for example, "If John is my brother, then we have the same mother and father." If the conclusion, that he is my brother, is not true, then the antecedent must be monotonically false. Things whose truth value is "default" true or false are believed to be true or false only in the absence of contradictory information.

So far as its eventual use is concerned, the project leaders hope that other researchers will build expert systems using the CycL language and the system's development capabilities, and thereby gain access to the Cyc KB and the benefits of the robustness of reasoning that they believe Cyc will eventually supply. Such projects would also extend the Cyc KB with new specialized but compatible information. The project leaders also hope that after enough information has been hand-coded into the system, it will have enough knowledge to be capable of substantial independent learning, say, through "reading" books or newspapers.

Before turning to more substantive comment on the project, it is perhaps useful at this point to try also to compare the scope of this project to some with which librarians are more familiar. In attempting to develop such a comparison in strictly quantitative terms, the author found herself to be very frustrated, since the various figures for the potential size of the system are quite inconsistently expressed in the few places in the book where they are described. However, she eventually stopped berating the authors and reminded herself that Cyc is a high-risk R&D project, not a contract to build a widget. In fairness to the authors, they do provide a specification of what functionality they want the system to have in 1994, which is far more important than exact size of the KB, no matter in what fashion one may choose to measure it.

However, some order-of-magnitude comparisons are possible. Lenat and Guha state in several places in their book that the KB must clearly contain at least millions of frames or their equivalent and tens of millions of pieces of data. The project also expects to devote two person-centuries to building the KB between now and 1994.

In comparing these numbers with, for instance, the *Dewey Decimal Classification* (DDC) (Dewey, 1989), the author made a rough estimate of the number of basic entries in that scheme (including the tables, but not the index or any synthesized numbers). This figure was in the neighborhood of only 30,000 to 40,000 entries, probably about 1/100th the number for frames contemplated for the KB.

While two person-centuries sounds like a lot of effort, a great deal more than that has been expended on the DDC over the years. One can grant that much of the expended effort has been in maintaining and redoing the scheme; developing it from scratch would be different. Still, the comparison does cause one to wonder about how much can be done, even with the amount of person power proposed. Not only will the rate of adding new entries probably deteriorate from whatever it is at present as the size of the KB grows, but anything being built over ten years' time will have serious maintenance problems before it is completed. These topics will be explored a bit further presently.

Hypotheses about Some Possibly Universal Problems of Large, General Knowledge Bases

One does not have to be a genius to develop a long list of problems such a project will have to solve: genius is required to solve them. Therefore, the exercise to be engaged in here of predicting some of these problems from the experience of librarians and commenting on whether they have arisen in the Cyc Project, and, if so, whether they have been successfully dealt with, is meant constructively and even humbly. The Cyc Project may or may not achieve what it is setting out to do, but its successes and failures will teach us a great deal. There is no way to experiment with large information systems at present except to build them. Moreover, the Cyc Project has an appealingly subversive character, not the least of which is that one suspects the project leaders are having fun. Spending one's days introspecting about why one does not believe certain articles in the *National Enquirer* might be quite addictive.

On a more serious note, however, there are at least four major barriers that have prevented a breakthrough in improving the effectiveness of large indexing languages and classification schemes beyond their present levels of utility. Since these barriers have arisen for general indexing and classification systems, they could certainly be expected to arise for this much more ambitious project.

The list below may seem strange at first glance because it is very general. There are hundreds of technical problems associated with Cyc, any one of which could generate pages of discussion and debate and any one of which could cause the project to founder. However, such debates are topics for the AI literature, not for this discussion. What the author is reacting to here is the statement that Lenat and Guha feel that the progress they have achieved in the past five years justifies raising their estimate of the feasibility of Cyc from 10-20 percent to 50-60 percent (p. 21). The four following points, listed somewhat facetiously, address why this author thinks the researchers may be overly optimistic based on the experience of librarians in constructing and maintaining large knowledge bases:

1. The variability of human performance in tasks related to KR and search or "My way, your way, and the Cyc way";
2. The paradox of structuring knowledge or Is more less?;
3. The double-edged nature of the 80/20 rule or The Law of Diminishing Returns; and
4. The inertia of the installed base or The Monster That Ate the Library of Congress.

Variability or My Way, Your Way, and the Cyc Way

Few facts have been more astonishing to information scientists or should give AI researchers more sleepless nights than the repeatedly demonstrated figures for indexer and searcher consistency, or rather inconsistency, in information systems. Much like the participants in several simultaneous games of gossip, a group of well-trained indexers or searchers can begin with the same text or request for information and emerge with less than 20 percent agreement in the outcome of their tasks, once the baseline information or the document being indexed has been conceptualized by the indexer or searcher and the concepts translated to fit within a formal structure.

The problem is not improving consistency. The main difficulty is that we do not know whether these low consistency rates are good or bad (Cooper, 1969). Inconsistency arising from error or complexity in rules, such as is being addressed by the Indexing Aid Project at the National Library of Medicine described by Humphrey in these proceedings, is, indeed, a worthy target for improvement, but clearly identifiable error accounts for a relatively small fraction of the variation. How can we improve consistency without reducing variety, in particular, variety related to linguistic expression, which is so much at the heart of human intelligent behavior? Or is it also desirable to reduce variety and if so, on what points? These questions are the truly hard ones for which we do not have any very good answers.

What do these observations mean for Cyc? First, of course, they raise grave questions about the degree of consistency that can be obtained in the Cyc knowledge-base development effort, even with a high degree of automated support. Lenat and Guha (1989) recognize the potential for inconsistency (p. 21), but one does not have the impression that they understand how large a problem it is. More troubling, however, is that until the KB is used, they probably will not know (1) how inconsistent the database is, or (2) what kind of problems the inconsistency will pose for them. The latter is the more interesting question, but since Cyc is a new sort of venture, it is difficult to speculate about it. Perhaps Cyc will be all too human: that is, it will produce useful results but have a high failure rate that cannot be self-diagnosed, particularly in the area of associations. The researchers are anxious for outside groups to make use of the Cyc KB, and it seems essential that this use be as early and vigorous as possible. Failure to exercise the KB as it is being built could produce some very unpleasant and expensive surprises.

The Paradox of Structure or Is More Less?

The principal paradox of structure is that it is simultaneously the essential ingredient and the primary barrier to the use of knowledge. At a personal level, everyone recognizes that the organization of his or her personal knowledge must be a key factor in the ability to exist in the world, to utilize sensory input, to interpret experiences and learn, or to use one's memory. Yet that very organization is a filter that can be a barrier to perceiving things that one should perceive or learning things that one needs to learn. A library shelving scheme, for instance, facilitates certain kinds of learning. Nonetheless, for all practical purposes, it prohibits others, presenting the user whose needs do not match its structure with something no better than a random ordering, at least in the worst case.

The broader and more unpredictable the use of a knowledge organization scheme, such as a general library classification, or even a subject database, such as ERIC, the more difficult it is for a high degree of organization to be universally helpful. This is a lesson that librarians think, at least, has been demonstrated even by comparative testing of retrieval systems. Some of the performance problems come just from the additional burden placed on the user by system complexity related to structure, which might be reduced or made less obvious through automated support. But much lies simply in the nature of knowledge, which is highly variable by culture and over time, and of information use, which filters it in many different ways.

How do the Cyc researchers expect to maintain the system's very highly structured and complex knowledge base? The project does have an answer for this problem, namely, that the system will become smart enough so that it can update itself with some coaching. If this cannot work, the Cyc researchers apparently would be among the first to recognize that continued maintenance by the same methods being used to build the KB would be untenable.

To take a topical example, librarians over the world are tearing their hair out considering how to update their systems to accommodate changes in European geography (an exercise they have gone through on several previous occasions during this century), but their problem is minimal beside that of updating the "common sense" information about world affairs, political systems, etc. that should eventually be contained in Cyc. If Cyc cannot read the newspapers, it is in real trouble—putting aside for the moment the problem of which newspapers it should read. In fact, one hopes the project members are saving their dailies, because the system is likely to be significantly out of date before they can get it built. Are they going to have to set a cutoff date and issue Cyc Circa July 1990 as a first release in 1994?

This author is perhaps even more bothered by the language and cultural problems. Although one of the project leaders is apparently an Asian immigrant, there does not appear to be any real appreciation that not everyone may want an embodiment (or is it an "enrulement") of a "1991 California/Texas Yuppie-Techie" as their consensus KB about the nature of the world. Yet no attempt is seen to bring in any broader perspectives. Aside from the temporal difficulties already discussed, one need consider only the probable analogs to the cultural bias that has caused such problems for librarians outside the United States (and sometimes for those in it) in using the DDC. One thing is certain, however: if Cyc is built, it will be an amazing artifact. Cultural historians will have a field day with it, at least if temporal snapshots of it are archived. Imagine having such a record of 18th century France or 16th century England!

The Double-Edge of the 80/20 Rule

The 80/20 rule, which has been repeatedly demonstrated to apply to the automation of things related to language, holds that algorithms or procedures can be found to handle 80 percent of the input with 20 percent of the effort. On the positive side, if one can identify and isolate (with a low error rate) the 80 percent of the cases for which the rules and procedures work well, a large percentage of the processing is susceptible to automation. However, the qualification to that statement is not trivial. It may be as difficult to throw an exception out of a system as it is to handle it correctly in the first place.

The negative side of the 80/20 rule, however, is that 80 percent of the effort covers only 20 percent of the cases, and this 20 percent causes the system to become vastly larger and more complex than the 80 percent rules would have led one to believe. As a friend once remarked from bitter experience, "When you have found the 80 percent algorithms, you have defined the problem." Consider, for instance, the *Anglo-American Cataloguing Rules* (Gorman & Winkler, 1988). The base rules occupy a few pages; the rest of this rather lengthy book is taken up with exceptions. Just to make things worse, as the data or KB grow in size, not only the absolute number but also probably the number of types of aberrant cases grows almost without limit. Thus, large KBs are inherently exponentially more complex than small ones, and such a system can never handle all cases. Some error must be tolerated. Related observations have been made by several other speakers at this conference who have addressed the law of diminishing returns in constructing a knowledge base.

The Cyc researchers mention some of these problems. In fact, their identification of the need for an ontology stems directly from the recognition that very large KBs are different. It answers the need they have identified to establish primitives in order to prohibit an infinite expansion of the database, much as librarians have attempted to do from time to time for the same reason. In addition, they have attempted to identify and address a range of problems before beginning any large-scale development in order to reduce backtracking.

Finally, they are attempting to encode information on a level of generality that would not bog them down in too much incidental detail. For instance, they would not record how to deal individually with the situation where a bag lady has scratched one's car with her cart vs. the situation where the owner of a Mercedes-Benz has rear-ended one's rattle trap. Cyc intends, instead, to record general principles about the right to try to recover when someone does damage to one's property and the notion that in order for someone to be able to give anyone something under any circumstances, they must have it—"it" in this case being money (Lenat & Guha, 1989, p. 22). This choice of level is directly related to the qualitative modeling problem previously discussed.

Nonetheless, this author is left with the nagging feeling that they have seriously underestimated the 80/20 problem. The book contains a great many descriptions of solutions that appear to be "80 percent algorithms." One has no way of knowing from the book how many other kinds of procedures have been incorporated into the system or what the researchers intend to do about handling exceptions, but there is cause to wonder. This area will be an interesting one to watch in future publications.

The Inertia of the Installed Base or the Monster that Ate the Library of Congress

Lastly, one of the problems with large systems is that they are large and one of the problems with systems that are used is that they are used. Both these sad facts of life tend to make it difficult to keep a large system up to date or make improvements to it, whether or not it is being updated automatically. This problem lies partially in the future for Cyc, but as soon as it contains a significant amount of data, design changes will become expensive. This fact is explicitly recognized by the Cyc researchers, as has been previously mentioned.

The conservative drag arising from the widespread use of systems is also well known to librarians, as it is to developers of commercial software and hardware. If Cyc is widely used as a component of other systems or as a host for them, the users will expect that updates will not seriously disrupt their own systems, which will probably inhibit major changes. Also, the project cannot wait too long before it acquires users. In order to test Cyc thoroughly, real systems need to be built with it, but it will be difficult to get developers to use it when it is only a laboratory product.

Clearly, we have another conundrum here, which suggests to this author that Cyc may become a test bed rather than a living system. As a test bed, it could have a vital role in expert system and natural language research even if the knowledge in it were frozen at a certain date or if areas of its knowledge were never completed.

The Cyc Project and Libraries

If the preceding remarks have sometimes sounded negative, this discussion of Cyc can close on a more positive but quite appropriate note by considering what a Cyc-like KB could do for libraries. Cyc in its projected 1994 form would have general knowledge of the world equivalent to that, say, of a high school student. It would be able to do some fairly sophisticated reasoning with that knowledge and would have at least a limited ability to learn from generally available external sources, such as textbooks or newspapers. If such a system existed, it might, for instance, be able to provide the basis for a natural language front-end for the *Sears List of Subject Headings* (Rovira & Reyes, 1986) that really could search and reason in a humanlike fashion. It would probably have a deep understanding of the vocabulary and concepts represented in a list of that size and generality, but not something with a high percentage of specialist terminology. However, with a basic ability to learn, it might be extended for special purpose uses.

It is sobering to think that the degree of effort represented by Cyc might be required to get a very intelligent information retrieval system even to this point of development, but it may be true. The gap between word-matching and deep understanding of language is a very large one and one that will probably only be bridged as part of large, cooperative development efforts in which libraries might serve as participants as well as beneficiaries.

CONCLUSION

While most papers in these proceedings focus on what AI can do for libraries, this one has attempted to show some of the many parallel problems between KR on a large scale and the problems of designing, developing, and maintaining large indexing languages and classification schemes. These connections are becoming better recognized. The teaching of AI in library schools is one such indication, and the recent founding of the International Society for Knowledge Organization by Ingetraut Dahlberg and others is also a step toward providing a forum for fruitful interchange of ideas between AI researchers and librarians. Indeed, the topics mentioned here are only some possible common interests. Just as many people trained in library science have become closely involved with data modeling and database design, many librarians could contribute to AI in general, whether through experience in building systems for libraries or through working on other applications. Both fields will benefit if the connections can be strengthened.

REFERENCES

Allen, J. (1989). Natural language understanding. In A. Barr, P. R. Cohen, & E. A. Feigenbaum (Eds.), *The handbook of artificial intelligence* (Vol. 4, pp. 193-239). Reading, MA: Addison-Wesley.

Buchanan, B. G., & Smith, R. G. (1989). Fundamentals of expert systems. In A. Barr, P. R. Cohen, & E. A. Feigenbaum (Eds.), *The handbook of artificial intelligence* (Vol. 4, pp. 149-192). Reading, MA: Addison-Wesley.

Clancey, W. J. (1989). Viewing knowledge bases as qualitative models. *IEEE Expert, 4*(2), 9-23.

Cooper, W. S. (1969). Is interindexer consistency a hobgoblin? *American Documentation, 20*(3), 268-278.

Dahlberg, I. (1988). Concept and definition theory. In *Classification theory in the computer age: Conversations across the disciplines* (Proceedings from the Conference, 18-19 November 1988, Albany, New York) (pp. 12-24). Albany, NY: Nelson A. Rockefeller College of Public Affairs and Policy, University at Albany, State University of New York.

Dewey, M. (1989). *The Dewey decimal classification and relative index* (20th ed.). Albany, NY: Forest Press.

Gorman, M., & Winkler, P. W. (Eds.). (1988). *Anglo-American cataloguing rules* (2d rev. ed.). Chicago, IL: American Library Association.

Humphrey, S. M., & Kapoor, A. (1988). *The MedIndEx system: Research on interactive knowledge-based indexing of the medical literature.* Bethesda, MD: National Library of Medicine. (LHNCBC 88-1, NTIS No. PB88-243902/AS)

Iwasaki, Y. (1989). Qualitative physics. In A. Barr, P. R. Cohen, & E. A. Feigenbaum (Eds.), *The handbook of artificial intelligence* (Vol. 4, pp. 323-413). Reading, MA: Addison-Wesley.

Lebowitz, M. (1980). *Generalization and memory in an integrated understanding system* (Research Report 186). New Haven, CT: Dept. of Computer Science, Yale University. (AD 093 083)

Lenat, D. B., & Guha, R. V. (1989). *Building large knowledge-based systems: Representation and inference in the Cyc Project.* Reading, MA: Addison-Wesley.

Quillian, M. R. (1969). The teachable language comprehender: A simulation program and theory of language. *Communications of the ACM, 12*(8), 459-476.

Rovira, C., & Reyes, C. (Eds.). (1986). *Sears List of Subject Headings* 13th ed. New York: Wilson.

Sharples, M.; Hogg, D.; Hutchison, C.; Torrance, S.; & Young, D. (1989). *Computers and thought: A practical introduction to artificial intelligence*. Cambridge, MA: MIT Press.

Shortliffe, E. H. (1976). *Computer-based medical consultation: MYCIN*. New York: Elsevier.

Walker, D. E. (1989). Developing lexical resources. In *Dictionaries for the Electronic Age: Proceedings of the 5th Annual Conference of the UW Centre for the New Oxford English Dictionary* (pp. 1-22). Waterloo, Ontario: University of Waterloo Centre for the New Oxford English Dictionary.

BRIAN C. VICKERY

Tome Associates, Ltd.
London, England

Intelligent Interfaces to Online Databases

ABSTRACT

The possible functions of intelligent interfaces are summarized. Five examples of recent or current European projects on the development of interfaces are described: INSERM, CIRT, EURISKO, ERLI/MINITEL, and IMIS. A number of the problems of interface design and implementation are reviewed.

INTRODUCTION

There has been an enormous investment in publicly available and corporate databases—bibliographic, numerical, directory, full-text, and so on. Despite all the aids provided by search services (text retrieval and database management systems), online access to databases remains difficult for many potential users. The user may need to know a variety of communications protocols, host command languages, search techniques, database file structures, and subject terminologies. In Europe, the natural language of the database may not be that of the user.

The aim of an intelligent interface is to make access easier by building some of the needed knowledge into front-end software used to interrogate the online search system. This aim does not coincide with that of creating an intelligent retrieval system. An interface accesses existing online systems, with all their constraints and deficiencies, so it can only be as successful as the online search system allows it to be. An interface does not address the problem of restructuring the database or the search system itself to make retrieval more intelligent.

In Europe, we are well aware of the pioneering work on intelligent retrieval that has been carried out in the United States—names such as Doszkocs and Croft come immediately to mind. But this article presents some of the work on intelligent interfaces that has been and is currently being carried out in Europe.

The Commission of the European Community has a division entitled "Telecommunications, Information Industries and Innovation"; a section of this has been particularly active in promoting interface work. During 1988-89, this section funded two "state-of-the-art" reviews by Cognitec (1988) and Vickery (1989). In 1989, it awarded two major contracts:

1. DISNET: an intelligent interface to online information to be implemented partly on a personal computer and partly on a host or network. The system will provide some general interface functions but also some specific to particular subject domains—agriculture and microbiology.
2. MITI: an intelligent multilingual interface (IMIS) on personal computers, to access a number of hosts. More will be said about this later in this paper.

FUNCTIONS OF AN INTELLIGENT INTERFACE

Online search of a database might be aided in a number of ways, aiming to:

- choose appropriate databases and hosts;
- permit the enquirer to state an information want in his/her own words;
- assist in clarifying the expression of the want;
- establish the level (introductory? advanced?) and approach (practical? theoretical?) of the information required;
- adjust the scope of the want (now become a query) so that the volume of retrievable information and the cost of the search are acceptable;
- formulate the query in the vocabulary used in the chosen databases;
- express the query as a search statement in the required format (e.g., using Boolean operators);
- handle the "housekeeping" activities of dialup, logon, file selection, downloading, and document ordering;
- transmit the search statement to the host using the appropriate command language, and, if necessary, switch between hosts and command languages;
- in search amendment, change the Boolean or other search operators,

and/or change search terms by various means, including relevance feedback; and

- present the search output in a helpful form, e.g., by ranking in order of probable relevance.

In a European context, a further important function is to aid the user whose natural language is not that of the database. Four levels of multilingual facility may be envisaged:

1. Multilanguage screen messages but query input in the language of the database(s) to be searched.
2. Input of search terms in one language and their immediate translation into the language of the database. (In these two options, if the user is not familiar with the language of the database, there can be little intelligent interaction with her/him in formulating and modifying the search strategy.)
3. Full processing of queries in more than one language, with translation of the final search statement into the language of the database. (In this option, full interaction can be achieved.)
4. Translation of search output into the language of the user. (This facility can in principle be added after any of the first three options, though it goes beyond interface functions into full-text translations.)

Following are five examples of European work on the development of interfaces exhibiting various degrees of "intelligence."

INSERM Interface

The French *Instit National de la Sante et de la Recherche Medicale* have developed an interface for searching MEDLINE on TELESYSTEMES-QUESTEL via the videotex system MINITEL (Halpern & Sargeant, 1988). The system uses the standard Minitel terminal and the user is prompted via menus.

A menu asks for entry of the major search criterion: French keyword, English keyword, English textwords, Author or Journal. When a criterion has been chosen, the user is prompted to enter a search term. If this is a textword, it is immediately searched. If it is a keyword, a listing of the MeSH keywords alphabetically surrounding the chosen term is displayed (in English or French), from which the user makes a selection. If the user initially enters a term that is a nonpreferred synonym in MeSH, then the preferred synonym and its alphabetical neighbors are displayed.

Once a keyword has been chosen from the alphabetical display, the user has the option of selecting more specific terms from the hierarchical MeSH thesaurus. The keywords one level lower in the hierarchy are displayed, from which the user may choose one; the process may be repeated down to lower levels until the user considers that his/her topic has been precisely expressed. The system now displays the MeSH subheadings that are valid for the search term, and the user is invited to select one or more of these.

From the keyword finally chosen, a subset is formed: the term is automatically ORed to all the more specific keywords derived from "exploding" that section of the MeSH hierarchy, the whole subset being linked to the chosen subheadings.

An initial search is now carried out. The number of references retrieved is displayed, and the user may either inspect them or narrow the search. To narrow the search, the user is presented with a further menu that asks for secondary search criteria, which may be of the same kind as the first or may be a limitation of the search. If the choice is to select a second criterion comparable to those in the first menu, the same procedure is followed as before, ending up with the choice of another search term. This term (or a subset derived from it) is then ANDed with the search based on the first criterion. This process of narrowing the search statement can be repeated.

Alternatively, a search may be limited to French language items or to clinical articles, or a search term may be required to be present in the "major keyword" field of a database record.

CIRT

CIRT is an experimental microcomputer front-end for searching certain databases, particularly MEDLINE, on the DATASTAR host (Robertson & Thompson, 1987). The user logs in to DATASTAR (using stored user i.d., password, and database name) and is then asked to specify limits (year, language, MEDLINE check tags such as human/animal, female/male, etc.). Subsequent interaction makes use of CIRT command language.

The user enters query terms which can be natural language words or MeSH terms. Any MeSH search facility can be used (e.g., explosion) or DATASTAR facility (e.g., truncation, adjacency). For example, suppose that three search terms A, B, C are entered. The system carries out the following searches:

1. A OR B OR C
2. A AND B
3. A AND B AND C
4. A AND B AND NOT C
5. A AND NOT B
6. A AND C AND NOT B
7. A AND NOT B AND NOT C
8. B AND NOT A
9. B AND C AND NOT A

10. B AND NOT A AND NOT C
11. C AND A
12. C AND NOT A AND NOT B

If a second-level search such as A AND B is reported by DATASTAR as retrieving nothing, then subsearches 3 and 4 are not made; the same would apply at searches 5 and 8. Weights are calculated for each of the third-level searches carried out, and up to fifteen records are downloaded from each set. The weight of each search term is inversely proportional to its postings in the database, and the weight for a retrieved set is the sum of the weights of its matching terms.

The retrieved sets are now ranked in decreasing weight order, and set details are presented to the user. The user can inspect items in each set in sequence from the (fifteen or less) items downloaded and, if desired, mark some items as relevant. When inspection of a particular set is completed, set weights are recalculated using a new estimate for term weight. A new term can be added with the effect that the necessary additional searches are carried out and set weights recalculated. A search term can be deleted, which has the effect of setting the term weight to zero. Individual items checked as relevant, or complete sets, can be selected for printing out record details offline.

EURISKO

This system has been implemented on microcomputer for an intelligent search interface, operating at present for searches of thirty databases on the TELESYSTEMES and CEDOCAR hosts (Barthes & Glize, 1988). The user is asked to choose a subject area of interest from a menu and to enter a subject query in French (English terms may also be used). A semantic grammar of fifty rules then analyzes the query, extracting data on the type of document requested, on any author name or language, and on the subset terms present. The system then tries to acquire further search-specific information from the user, asking, for example:

—If you wish to truncate "dyadic," give me the root.
—How many characters should be sought after truncation?
—Is the word "and" in "dyadic functions and piezoelectricity" a link between two concepts (y/n)?

Based on the subject area of interest and on the type of document requested, a list of databases is displayed to the user in order of probable relevance. Several databases may be selected by the user. Connection to the selected hosts and databases is automatically established, and the system prepares to transmit a search request.

A query on "chemical composition of the leaf of sweet corns and of the stalk of sorgho" would be analyzed into components which are tagged as follows.

	root	truncation	operator
1 -	chemical	N	PROX
2 -	composition	N	AND
3 -	lea???	Y	AND
4 -	sweet	N	PROX
5 -	corn?	Y	OR
6 -	stalk?	Y	PROX
7 -	sorgho	N	NIL

The operators indicate the relation of a term to the following term in the sequence. The PROX operator implies that a proximity operator will be needed between the two terms. The search terms and appropriate operators are transmitted to the host system one by one under the control of a set of sixty rules. In this case, the following search statements would be generated, and the number of postings would be returned by the host at each stage:

1 - chemical PROX composition
2 - lea???
3 - sweet PROX corn?
4 - 2 AND 3
5 - stalk?
6 - sorgho
7 - 5 AND 6
8 - 4 OR 7
9 - 1 AND 8

At each step, errors can be recognized that need correction. For example, at any step the number of postings might be zero. Rules control the actions that the system then takes, for example, to ask the user for a synonym of a zero-posted term, which is then used for search. If the overall search retrieves zero postings, the search must be broadened in consultation with the user.

When the search has retrieved some items, these are displayed to the user for a relevance judgment, which may be that the results are too general, or too specific, or relevant but insufficient, or off-focus. Appropriate action is then taken. For example, to narrow a search, the system interrogates the user in turn about:

- amending truncation to be more specific,
- eliminating ORed terms,
- adding new ANDed terms,
- altering the operators, e.g., changing AND to PROX,

- restricting search to named fields, or to specified dates, or to types of documents, or to language.

After each amendment, a fresh search and evaluation takes place.

ERLI/MINITEL

The firm ERLI has developed an interface via MINITEL terminals to the professional headings of the French Yellow Pages directory (Clemencin, 1988). The system naturally uses French, but in the description below, the examples will be mainly in English.

The Yellow Pages are normally accessed through about 2,500 headings, e.g., lampshades (manufacturing and trade), estate managers and co-ownership trustees, rubber products for sanitary use (manufacturing); domestic vacuum cleaners and floor polishers; or typewriter, accounting and invoicing machine hire.

Headings may have subheadings chosen from a standard set or assigned by the agency listed in the directory. The technique normally used to access the directory is by keywords: a user query such as "I would like to book seats on a holiday tour" is analyzed to eliminate "empty" words and a Boolean expression is created: AND (book, holiday, seat, tour). This is then used to search for headings containing the ANDed words. If no output is obtained, the expression OR (book, holiday, seat, tour) is searched; this all too often results in a match with many headings.

The ERLI interface differs in two ways: the headings are indexed as described below and are approached via the index, not directly, and queries are handled by language processor. A study of the headings used in the Yellow Pages indicated that they contained three kinds of words:

1. so-called "predicates" expressing the activity of an agency in the directory, e.g., sales, manufacture, repair, hire, retail;
2. "empty" words such as supplies, equipment, contractor; and
3. "primary" words—the main bulk of words referring to objects such as furniture, cars, etc., or names of professions such as printer, surgeon, architect.

The index contains 20,000 entries. Each entry consists of a single or compound word. Rules allow the recognition of a word through all its inflectional variants (e.g., *social, sociale, sociales, sociaux; sport, sportif; fabriquer, fabrication*). Compounds may be of various kinds, e.g., *salle de bain, pomme de terre, train electrique (miniature)*.

To each entry is attached a grammatical category, links to terms that are semantically related, and pointers to the headings which it indexes. Each heading is indexed either by its primary term alone or by a compound of an empty word and a primary term. Predicates are indicated in the form of a relation between the index entry and the heading; there are twelve such relations, their English equivalents being: retail, wholesale, manufacture, repair, renting, transport, design, medical care, reservation, lessons, training, and custom-made contracts. An entry may be linked via several predicate relations to a number of headings. The semantic links are to synonyms and to broader terms.

Analysis of user input begins by recognition of single words, and each word is looked up, in turn, in the index. If it occurs more than once (i.e., it is ambiguous), rules are invoked that take context into account to resolve the ambiguity. If the word as entered is not in the index, stemming rules derive a standard form and variants of this are sought. If still not found, a spelling correction procedure is invoked that creates a phonological representation of the word; this is compared with the phonological representations of single index words. If still not identified, the word is treated as unknown. Compounds occurring in the input are identified next. Terms which are synonyms for a preferred term (as used in headings) are replaced by the heading terms.

The treated input is now processed by sets of grammatical rules, which identify elements as conjunctions, standard subheadings, predicates, and primary terms, and take appropriate actions to transform the input into a query that can be matched against Yellow Pages headings. An example of this process is the treatment of the input query (here given in English):

> "steel rim for car wheel"

The string contains only primary terms. It does not occur as it stands in the index, although the individual words are known. The system tries to generate variants by using broader terms:

> "steel accessory for car"
> "steel rim for wheel of vehicle"
> "metal rim for car wheel"

These are not found in the index so the system simplifies the input by dropping terms, to give the searches:

> "car wheel"
> "steel rim"

and then again tries broader terms, achieving an index match with:

> "car accessory"

which points to a Yellow Pages heading:

> automobiles (detached components and accessories).

This is searched together with the other original primary terms ("rim" and "steel") as subheadings.

IMIS

This year the European Commission awarded a contract to a consortium to develop an intelligent multilingual interface to databases, mounted on an IBM PC and accessing, in the first instance, a number of European hosts. The consortium partners include Tome Associates (UK) who have previously developed the commercial software TOME SEARCHER and TOME SELECTOR; the University Paul Sabatier, developers of EURISKO; and Softex GmbH, a German firm specializing in multilingual text processing. A new interface is to be constructed, building on the products and techniques already existing among the partners.

The functional scope of the proposed interface can be seen from the figure titled IMIS in Action. The user will be able to choose one of four languages in which to interact with the system: English, French, German, and Spanish. She/he will then be asked to indicate the general subject area of the query—if necessary, being guided down a hierarchical menu of subjects. The system will display descriptions of databases in the chosen subject area, and the user will select one or more of these and, if necessary, the preferred host. This part of the new interface will be based on the existing TOME SELECTOR.

At this point, several alternatives will be available:

1. It may be that the user wishes to access a host "not known to IMIS." The system will then be used simply as a communications package, and the user him/herself must dial up, provide identifier and passwords, logon, and carry out a normal search.
2. The host may be "known to IMIS" but the user does not want "aided search." In this case, the system will provide automatic dial-up and logon, but the user must input a search using the command language, Boolean operators, and other search techniques of the chosen host.
3. The user wants "aided search" but in a subject area not covered by the IMIS dictionaries. She will in this case be guided in query development along the "user-based" path: essentially, the system will use the procedures described in EURISKO.
4. The user wants "aided search" in a subject area covered by the IMIS dictionaries (the subject areas to be covered are technology in general and environmental information). In these areas, the user will be guided along the "thesaurus-based" path, which will be a development of the existing TOME SEARCHER procedures now to be described (Vickery, 1988).

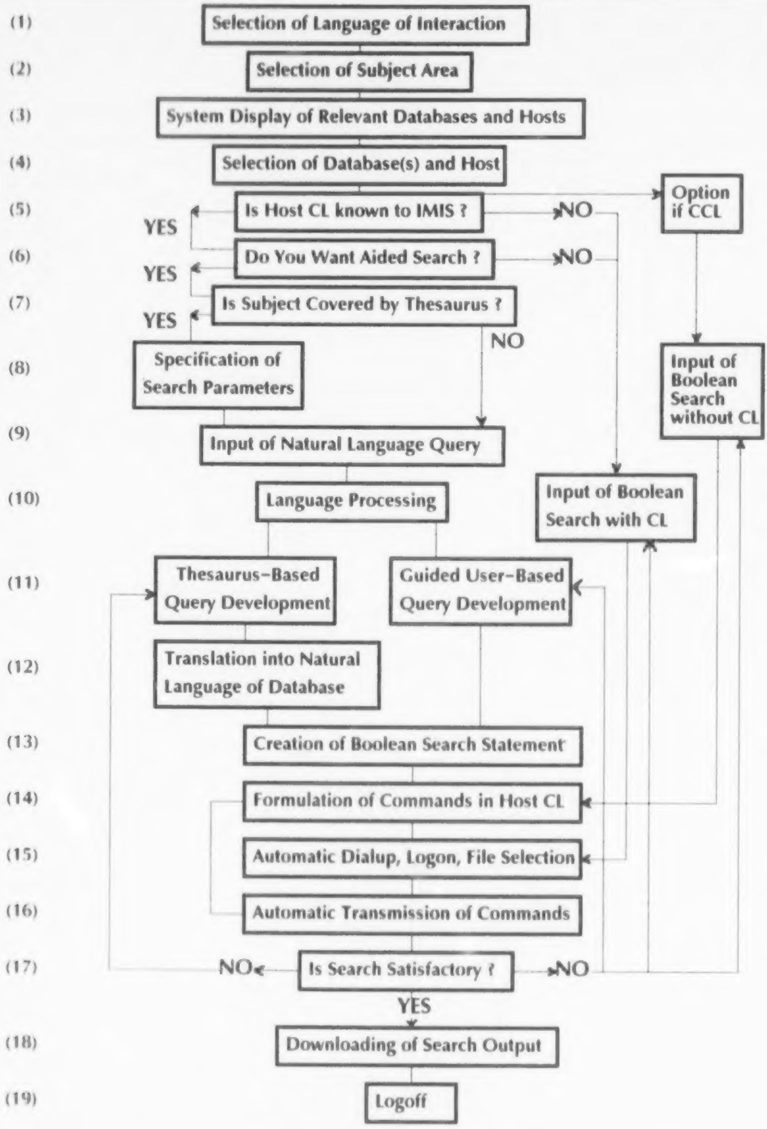
In "thesaurus-based query development" the user is asked to specify various "search parameters," e.g., Is the search to be by author or subject? Should it be precise? Should it be limited by date, language, or document type? What output format is required? How many output items should the search aim to produce? He/she then inputs a natural language query. Automatic language processing includes:

- separating the input into words;
- checking each word against an extensive stoplist;
- stemming each word not stopped;
- checking each stem against a dictionary (because stems with more than one meaning occur more than once in the dictionary, and for each stem there is recorded: a semantic category, a pointer to a position in a subject classification, and a pointer to any synonyms);
- clarifying the meaning of any stems not in the dictionary by interaction with the user (checking the spelling, assigning the word to a semantic category, and locating its position in the subject classification);
- checking successive stems in the input to see if they occur as compound phrases in the dictionary;
- disambiguating any remaining multimeaning stems;
- forming new compounds (not in the dictionary) from remaining successive stems in the input using rules on permissible combinations of semantic categories; and
- recognizing indicators of negation in the input.

Since the user is able to interact with the system in any one of four languages, these language-processing facilities have to be provided in each of those languages. If the search statement has been constructed in a language different from that of the chosen database, automatic procedures to translate search terms between English, French, German, and Spanish will be provided. A Boolean search statement in the language of the database will then be automatically constructed and transmitted to the host using the appropriate command language, and the search results will be automatically downloaded. If the initial search is not satisfactory, the system will return to an earlier stage to use thesaurus assistance in reformulating the search.

PROBLEMS OF INTERFACE CONSTRUCTION

The IMIS team is just completing its detailed design document, but we would be far from claiming that the problems of interface design have been resolved. Let us consider some of them.


(1) Selection of Language of Interaction
(2) Selection of Subject Area
(3) System Display of Relevant Databases and Hosts
(4) Selection of Database(s) and Host
Option if CCL
(5) Is Host CL known to IMIS ?
YES
NO
(6) Do You Want Aided Search ?
YES
NO
(7) Is Subject Covered by Thesaurus ?
YES
NO
(8) Specification of Search Parameters
Input of Boolean Search without CL
(9) Input of Natural Language Query
(10) Language Processing
Input of Boolean Search with CL
(11) Thesaurus-Based Query Development
Guided User-Based Query Development
(12) Translation into Natural Language of Database
(13) Creation of Boolean Search Statement
(14) Formulation of Commands in Host CL
(15) Automatic Dialup, Logon, File Selection
(16) Automatic Transmission of Commands
(17) Is Search Satisfactory ?
NO
NO
YES
(18) Downloading of Search Output
(19) Logoff

IMIS in action

Analysis of Natural Language Query Input

To allow a user to express a query in her/his own words seems to be a very necessary feature for an interface that aims to be "intelligent." Natural language processing (NLP) is itself an art still under development. Queries display only a subset of natural language structures (e.g., they rarely contain verbs) and so are simpler syntactically. But analysis has to transform them into structures that represent the semantics of search statements, and this involves problems not always handled by conventional NLP. Here are a few of the issues:

- At the simplest level, how should misspelled words be recognized, and, if recognized, how handled? Should the system attempt to correct spelling?
- How will the system handle hyphenated words?
- There are unresolved problems in the recognition of compound terms. For example, how can we avoid forming a noun combination "cat food" in a sentence such as "It is necessary to give the cat food?" How is a long noun phrase to be broken up, e.g., "airport long term car park vehicle pickup point?"
- There are stock phrases and idioms such as "other things being equal"; how should they be recognized and handled?
- In some subject domains, there are also specialist phrase structures, e.g., dates such as "Monday March 24, 1989" that need special treatment.
- In general, how are numerals to be dealt with, such as "24 volts" or "Boeing 747?"
- How should enumerations be handled, e.g., "smog pollution control" (is "smog control" under discussion) or "hard and floppy disks" (should we form the phrase "hard disks")?
- Ellipsis is the practice of leaving out some data in a text string because it can be inferred from the context. A simple example is "the melting point of sulphur and the boiling point" ("of sulphur" is not explicitly stated). How will this be recognized and handled?
- Will it be necessary to handle pronoun reference? For example, what does "their" refer to in the following expressions?

  —the colors of dyestuffs and their chemical structures
  —the colors of dyestuffs and their fading
- Will the system have to cope both with grammatically well-formed sentences and with sentence fragments or ungrammatical input?
- Will the input make consistent use of capitals (for proper names, acronyms, etc.) so that they can be used to aid analysis?
- If a stoplist is used, how will the system handle a homonym such as the stopped word "and" and the expression "AND logic?"
- If a Boolean search statement is to be created, how will the system

> know when to link search terms by AND, when by OR, when by AND NOT? For example, how should a query such as "Comparison of statistical and linguistic methods of indexing and abstracting" be represented as a search statement?

It is very difficult to construct a robust language processor that sensibly handles all types of user query.

Lexicons

By "lexicon" is meant *any kind of word file held within the interface system.* IMIS will contain monolingual dictionaries in four languages, pointers between language pairs, and pointers between words with thesaural associations (synonyms, broader and narrower terms). The creation of large lexicons presents many intellectual problems and is very labor intensive.

Subject Scope

Much experimental work on interfaces to information systems has been carried out within narrow subject limits. This clearly also limits the range of application of an interface and hence the number of potential users. For an interface to be commercially viable, it will have to handle a wide subject scope. This immediately increases the problems presented by lexicons; in particular, the problem of ambiguity—words with multiple meanings. There are few standard ways of resolving an ambiguity. A specific rule for each particular word must be constructed in most cases, and such rules are rarely foolproof. In a system of wide scope, one is no longer working with a subject domain that has a clearly defined semantic structure. It becomes more difficult for an interface to transform a query into a unique semantic representation that can be used in a search statement.

Query Modification

This includes both the process of clarifying and adjusting a query before search, and the process of revising a query if first search results are not satisfactory. There are two aspects to query modification. First, what ways of amending a query are open to intelligent interface? Second, what is the best balance between man and machine, i.e., should the interface make modifications automatically or should it simply advise the user as to what modifications are possible and leave him/her to take action? Also, an interface cannot make a general recasting of a query. It can only operate in small, discrete steps such as:

- adding a term to a query (using AND, OR, AND NOT),
- removing a term from a query,
- replacing one term with another,
- altering a Boolean operator (e.g., AND to OR),
- altering a term (e.g., by truncation), or
- altering a search limitation (e.g., field, date, language).

Making or suggesting changes a step at a time can irritate the searcher. The sequence of changes offered may not be acceptable. The procedures cannot easily cope with the user who suddenly has an insight into the query she or he should have put. The problem is learning how to provide guidance while retaining flexibility.

The degree of user initiative offered will reflect the views of the interface designer on how capable the user is of making search decisions. The ERLI/MINITEL system described earlier was explicitly designed on the assumption that the bulk of users could not make effective use of the subject headings of the French Yellow Pages. The EURISKO systems expects the user to be able to supply the terminology of the chosen subject, offering guidance only as to the kind of actions that may profitably be undertaken. IMIS plans to offer a variety of alternative degrees of user involvement.

Interface and Database

The interfaces described in this paper have all been situated with the user, and the software incorporated in a microcomputer. In this situation, there arise the issues: How much search preparation can the interface provide before going online? Can the interface continue to provide search aid when the user is already connected with the database?

Alternative ways of providing intelligent search aid are:

1. to mount an interface on a gateway node in the telecommunications network, accessible from each user's terminal: the DISNET project mentioned earlier will be exploring this possibility;
2. to mount it within a host computer: the European Commission is funding work on its own host, ECHO, using this configuration; or
3. to mount it in a microcomputer that also contains software to search local CD-ROM. This is a configuration that needs to be actively explored, especially as the possibility then arises of the interface software making active use of the indexes and thesauri stored on the CD-ROM.

CONCLUSION

This paper has tried to present some of the achievements, possibilities, and problems of constructing intelligent interfaces to online databases, arising out of European experience. Despite the effort that has gone into—and is continuing to go into—the development of practical systems, there is still a feeling in Europe that more analysis of the problems and experimentation with possible solutions are needed. This feeling is reflected in the existence of another European Commission project in which Tome Associates is involved, a project known as SAINT: Simplification of Access to Information using Normalised Transfer. The project is designed to collect further information on interface design, to come up with a more refined modular architecture, and to suggest experiments for the testing of particular modules.

REFERENCES

Barthes, C., & Glize, P. (1988). A case study of planning in information retrieval. *Expert Systems for Information Management, 1*(1), 50-65.

Clemencin, G. (1988). Querying the French Yellow Pages: Natural language access to the directory. *Information Processing & Management, 24*(6), 633-649.

Cognitec. (1988). *Simplification of access to information using normalised transfer: State-of-the-art analysis.* Report presented to the Commission of the European Communities.

Halpern, J.; Sargeant, H. A.; & IMA Centre de Documentation de l'INSERM, France. (1988). A new end-user interface for bilingual searching of MEDLINE. In *Online Information 88: 12th International Online Information Meeting* (pp. 427-443). Oxford: Learned Information.

Robertson, S. E., & Thompson, C. L. (1987). *An operational evaluation of weighting, ranking and relevance feedback via a front-end system.* London: City University, Department of Information Science.

Vickery, A. (1988). The experience of building expert search systems. In *Online Information 88: 12th International Online Information Meeting* (pp. 301-313). Oxford: Learned Information.

Vickery, B. C. (1989). *Intelligent interfaces for user-friendly access to databases: State-of-the-art survey.* Report presented to the Commission of the European Communities.

Vickery, B. C., & Vickery, A. (1990). Intelligence and information systems. *Journal of Information Science: Principles & Practice, 16*(1), 65-70.

JAIME PONTIGO
EZEQUIEL TOVAR-REYES
GUILLERMO RODRIGUEZ
SERGIO ORTIZ-GAMA

Instituto de Investigaciones Electricas
Cuernavaca, Mexico

Expert Systems in Document Delivery: The Feasibility of Learning Capabilities

ABSTRACT

To solve the problem of document delivery in Mexico, the authors developed SEADO (Expert System for Document Supply). SEADO consists of three main components: a knowledge base, an expert system shell, and the database. The knowledge base was built through fault tree analysis and through structured flowcharts. The shell was developed with EXSYS, a generalized expert system development package. The database was based on information sources of various kinds: printed material, local databases, public databases, etc. To evaluate the impact of different learning capabilities, the authors decided to test alternative ways of achieving a predictor for the system to perform in a dynamic and adaptive way. Learning by a weighted-based scheme was compared with a probability-based scheme.

INTRODUCTION

Today, to be able to get a surrogate from a foreign database is almost trivial, but getting one's hands on a document can be more or less cumbersome at different latitudes. The problems involved in document delivery do not seem to be of great concern to the builders of expert systems (ES). A recent search on the literature of this subject reported only two efforts in this direction (Bianchi & Giorgi, 1986; Waldstein, 1986). The authors have good reasons to believe that this application is the first of its kind in Mexico as well as in all of Latin America.

The first part of this paper explores the conditions where SEADO (Expert System for Document Supply) was conceptualized; the second is devoted to the architecture of SEADO. After some background, the last part deals with the control sketch topic: learning capabilities.

EXPERT SYSTEM FOR DOCUMENT SUPPLY (SEADO)

SEADO has been under consideration for some time as a way of achieving several goals that have remained unfulfilled due mainly to lack of human resources in the area of librarianship in Mexico. Briefly, this paper will explain the reasons behind trying an ES as an alternative way to solve some problems. Table 1 sketches the current environment where the expert system is designed.

TABLE 1
THE WORKING ENVIRONMENT AT THE TECHNOLOGICAL INFORMATION NETWORK FOR MEXICAN UTILITIES

Population served:	690 Researchers
	4,000 Engineers
Means:	Network of 13 special libraries in electric utilities and industry's R & D labs
Acquisitions:	5,000 requests unfilled annually
Types:	30% Journal articles
	25% Conference proceedings and books
	15% Conference papers
	13% Technical reports
	9% Patents
	8% Standards
Constraints:	Incomplete collections (locally and nationally)
	Lack of funds for acquisitions
	Lack of trained staff
	Pressure for expediting
	Poor telecommunications network
	Lack of understanding of the importance of library

Why Should We Start an Expert System?

Apart from the long list of problem criteria given by Liebowitz and DeSalvo (1989, pp. 6-8) which for the most part holds, the authors wished to pursue the following goals:

1. capture the experience from the experts available;
2. make better distribution of human resources;
3. help in making better decisions (thus saving time and money);

4. free the experts from routine tasks;
5. ensure continuous operation in the absence of the experts; and
6. improve the quality of library operations.

The expected results in terms of day-to-day operations should be:

- The expedition of pre-ordering searching
- The evaluation of the best supplier
- The expert's support in decision making
- The expert's knowledge upgrade

SEADO Architecture

Liebowitz and DeSalvo (1989) have defined the process of expert systems construction as follows:

> Building an expert system is an incremental activity which involves the development, critiquing, and subsequent refinement of a succession of prototypes. The successive approximation of the final expert system depends on the results of user trials with the prototypes. (p. 38)

An important aspect in the development of the expert system is the design of its structure or architecture. As Hayes-Roth et al. (1983) have established, the term *architecture* refers to the science and method of design that determine the structure of the expert system. The emergent principles reflect current understanding of the best way to design structures that support intelligent problem-solving. In this context, the architecture of the SEADO consists of the following main components: A knowledge base (KB), an expert system shell (ES), and the database (DB). These components are described briefly below.

The Knowledge Base (KB)

The real power of an expert system is the knowledge base, since it contains the available knowledge of the human experts which is generally developed by the interaction of a knowledge engineer and the knowledge expert in the domain of expertise.

Various methods have been proposed to acquire and formalize knowledge concerning a special universe of discourse (Chachko & Stakbovaya, 1972; Eick & Lockemann, 1985; Weiss & Kulikowski, 1984; Yung-Choa Pan, 1984). Tools from conventional systems analysis can improve the knowledge engineering process through formalization and standardization of expert systems building methods. One of the major advantages of this approach is that it produces a set of specifications, explicative and graphic, for the empirical performance of the system. Knowledge engineering is, after all, a creative science wherein can be developed systems that imitate the behavior of a human expert even though the underlying computer system is vastly different from the human mind in its form, functions, and capabilities (Liebowitz & DeSalvo, 1989, p. 64).

At Instituto de Investigaciones Electricas (IIE), the authors have been using successfully two methods for knowledge acquisition: Fault Tree Analysis (FTA) and Structured Flow Charts (SFC) (Rodriguez & Rivera, 1986).

The FTA approach to building KBs is especially suitable when knowledge is presented in the form of engineering drawings, operational guidelines, maintenance procedures, and heuristic rules. The SFC approach is more adequate when knowledge is procedural and is obtained directly from human experts or from a manual or handbook.

Thus, when the SFC approach is used to build KBs, the charts explain how the human expert makes decisions and arrives at conclusions. If the flowcharts come unstructured from the expert, they should be structured by using only the basic building figures of structured flowcharting: the sequence, the decision, and the loop or cycle (McGowan & Kelly, 1976).

The knowledge base consists of representing human expert knowledge in the form of an SFC which is easily converted to production rules. Figure 1 shows how rules are obtained for each one of three basic structured figures. These rules are condition-action pairs which specify that IF some condition is true, THEN some action is performed.

Production rules, like a knowledge representation technique, have the following advantages:

1. They are easy to express, to understand, and to work with.
2. Every rule expresses a decision procedure.

The rules obtained to select the appropriate supplier using the SFC approach have been divided into seven groups, one for each type of document request: books, conference papers, conference proceedings, journal articles, technical reports, and standards and patents.

In the case of books, there are six possibilities for the assignment of a supplier when the place of publication is Mexico. The place of publication is established via the breakdown of information in the ISBN table. The information about this table that is used by the system will be described later. Figure 2 shows the SFC for the latter case.

Some of the rules obtained from the flowchart follow:

IF publisher from Mexico and book at Gonzalez Libros Tecnicos (GLT)
THEN order to GLT.
IF publisher from Mexico and book not at GLT and is found in Table A-1 and book at American Bookstore (AB)
THEN order to AB.

IF publisher from Mexico and book not at GLT, not at AB and is found in Table A-2 and Book at Delti
THEN order to Delti.
IF publisher from Mexico and book not at GLT, not at AB, not at Delti and is found at LL
THEN order to Local Libraries (LL).
IF publisher from Mexico and book not found at GLT, AB, Delti and LL
THEN order to publisher.

These rules have been captured and stored in a generalized expert system development package which is described below.

Inference Machine (Shell EXSYS)

At the beginning of SEADO's development, several expert systems' programming languages were considered in the design of the KB and the Inference Machine (IM). Recently, the shell EXSYS was selected because it seems to have advantages over other programming languages. Some of these advantages are shortened ES development time, more facilities such as an input processor, and explanation mechanism, and a rule tracer which debugs the KB. Furthermore, the shell EXSYS is more suitable to the authors' needs since the expert's knowledge is easily represented as in production rules.

EXSYS is a generalized expert system development package which asks the user questions relevant to a subject, and has the user answer by selecting one or more answers from a list or by entering data. The computer continues to ask questions until it reaches a conclusion. This conclusion may be the selection of a single solution or a list of possible solutions arranged in order of likelihood. The ES can explain how it arrived at its conclusion and why.

The development of the ES with EXSYS can be applied to any problem that involves a selection among a definable group of choices where the decision is based on logical rules. Furthermore, the rules can involve relative probabilities or weights (certainty factor) of a choice being correct.

EXSYS can communicate with external programs for data acquisition, calculation or result display, and data can be passed back to EXSYS for analysis. Furthermore, EXSYS can receive data directly from databases and spreadsheets.

The Database (DB)

The database which provides the necessary data that the expert system uses to execute some of the rules associated with it contains tables (dictionaries) based on information sources of various natures, i.e., printed repertories, local databases, public databases, etc.

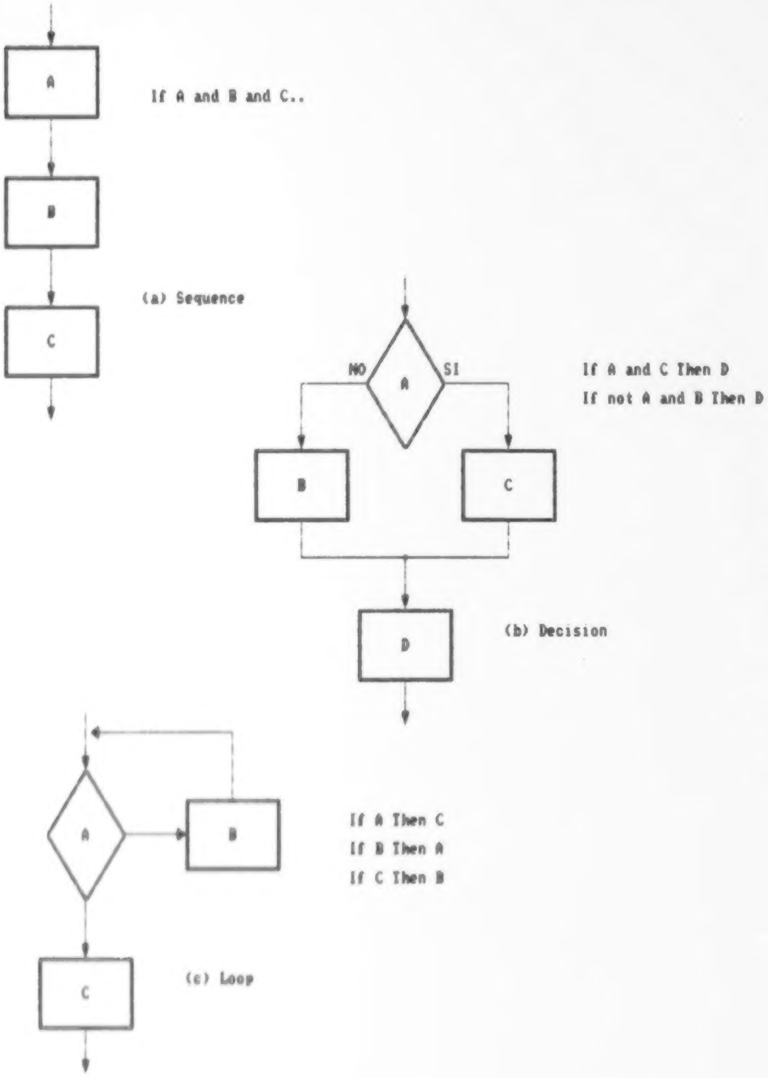

A
If A and B and C..
B
(a) Sequence
C
NO
A
SI
If A and C Then D
If not A and B Then D
B
C
D
(b) Decision
A
B
If A Then C
If B Then A
If C Then B
C
(c) Loop

Figure 1. Conversion to production rules of the three basic building blocks of structured flowcharts

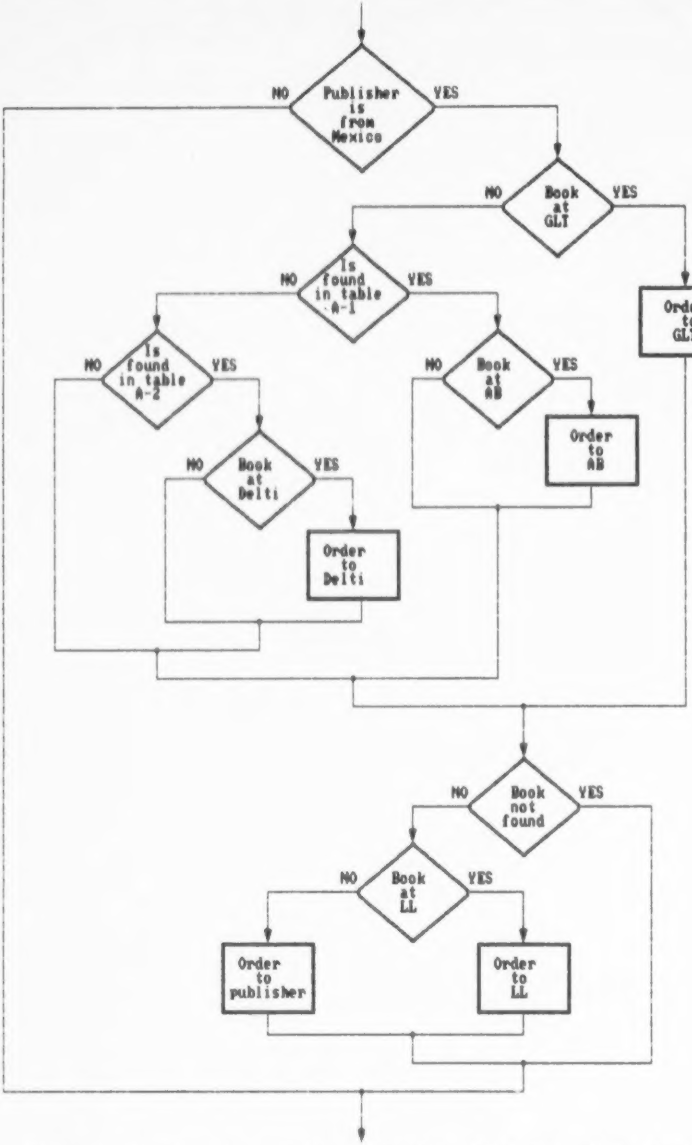

Publisher is from Mexico
NO
YES
Book at GLT
NO
YES
Is found in table A-1
NO
YES
Order to GLT
Is found in table A-2
NO
YES
Book at AB
NO
YES
Order to AB
Book at Delti
NO
YES
Order to Delti
Book not found
NO
YES
Book at LL
NO
YES
Order to publisher
Order to LL

Figure 2. Flowchart for suppliers when the publishers are from Mexico

For each type of bibliographic material, there exists a set of tables that the expert system uses to identify certain parameters which allow the ES to select the supplier. Here only one of the tables and its main function are described. Details on the tables built for this purpose are given in Pontigo et al. (in press).

In cases where the bibliographic material contains the ISBN number as a data element—for example, books and conference proceedings—Table 2 includes a list of ISBN numbers, places of printing, and publishers.

By means of this table, the expert system finds and identifies data (such as publisher) that some of the rules request to be fired. The database can be enriched at any moment with relevant information which will be evaluated periodically to ascertain its value for the system.

TABLE 2
LIST OF PUBLISHERS (EXAMPLE)

ISBN	*Place*	*Publisher*
0-387-10771-1	New York, NY	Springer-Verlag
0-12-685480-7	Orlando, FL	Academic Press
0-07-023655-0	New York, NY	McGraw-Hill

LEARNING CAPABILITIES

The reasons for exploring these attributes have been given elsewhere in regard to the expert system at large. In this case, the number of occurrences is not big enough to be considered in any way the only reason for this endeavor, but it is sufficient to illustrate the benefits obtained on a larger scale.

The learning capabilities of SEADO are geared to maintain size control and, at the same time, to streamline the system and allow for a more efficient overall operation.

Previously mentioned was the broad problem to be solved with the expert system—namely, to make the operation of document delivery more efficient through the use of prior experience and ordering information. The problem may be stated more precisely as follows: *We need to design an expert system for document delivery that is dynamic and adaptive.* Adaptivity and dynamism are attributes which were not simple to achieve mainly because of the amount of information that has to be dealt with in traditional systems. No doubt, many others have come across these problems, but there are no direct references to them in the literature.

The authors have raised the following questions related to learning capabilities in regard to the design of ES, both in dynamic and adaptive behaviors:

1. How can we design in order to guarantee the best use of information used in the process?
2. There have to be changes in the system with the acquisition of a particular type of material. What changes? How much change?
3. What number of cases processed yearly is significant for depletion rules to hold without degrading the quality of decisions?

The Experiment

In order to evaluate the impact of different learning capabilities, the authors decided to test alternative ways of achieving a predictor for the system to perform in a dynamic and adaptive way. Learning via a weighted-based scheme was compared with a probability-based approach.

The design of the ES incorporated a criterion for depletion rules based on Pareto's Law of Diminishing Returns, also known as the 80/20 rule. According to Pareto, 80 percent of the orders should be delivered by 20 percent of the suppliers. The expectation is that the databases used as sources for the experts' decisions be streamlined with the same rule at least once a year. The idea was to compare one of the weighted criteria with another, based on the probability of acquiring something given prior acquisitions history.

The best data available for the first comparison used the criterion "Potential use in research projects" as expressed by the population of the originating sources as represented in the shelflist, to be compared with data available on acquisitions from 1985 to 1989 (see Table 3).

Using information about the suppliers from each group described, data were ranked and compared using the Spearman Correlation Coefficient (rs) with results of $rs = 0.607$ when all suppliers were included and $rs = 0.19$ when the two biggest suppliers were excluded. This shows the poor correlation of the two criteria used. Figures 3 and 4 show the cumulative distribution in both cases.

In the comparison of probabilities, the probability of the report's producer being a contributor was rank-correlated both to the acquisitions from 1985-86 and to 1986-89. The Spearman Correlation Coefficient (rs) finding was $rs = 0.922$, with very little distortion when the two big suppliers' data were pulled off: $r = 0.858$.

The technical reports purchased come mainly from two suppliers; however, thirty-six sources have been used from 1985 to 1989. Table 3 shows the participation of the suppliers.

It is sound to suppose that, for the expert system, it is simple to discriminate the data supplied regardless of the degree of participation of the supplier. In fact, the identical data source used for the selection of suppliers allows for alternative ways of becoming more efficient. The procedure would be to branch before the 20 percent of suppliers is defined. With two big suppliers, namely, NTIS and EPRI (6 percent) providing 86 percent of the reports, those two can be channeled before looking at the table for other suppliers, thus providing the opportunity to apply the 80/20 rule over the 14 percent left. In this way, the selection can be achieved over 85 percent, plus 80 percent of the 14 percent, for a total of 97.2 percent.

TABLE 3
DEALER'S ORDERS, TECHNICAL REPORTS (1985-1989)

	Code	*'85-'86*	*'87-'89*	*Total*	*Prob. (%)*
1	AAP		1	1	0.09
2	ABH		1	1	0.09
3	AO	1		1	0.09
4	BAE		1	1	0.09
5	BAI	5	4	9	0.88
6	CAW		1	1	0.09
7	CAZ	2	5	7	0.68
8	CBK	1		1	0.09
9	CBX	4		4	0.39
10	CBY	3	1	4	0.39
11	CCA	5	2	7	0.68
12	CCM	4	4	8	0.78
13	DBA		5	5	0.48
14	EAO	124	156	280	27.31
15	EAQ	9	11	20	1.95
16	EBF		5	5	0.48
17	EBP	4		4	0.39
18	ECE		2	2	0.19
19	ECG		1	1	0.09
20	ECI		1	1	0.09
21	GAC		7	7	0.68
22	GAE		1	1	0.09
23	GAY	2		2	0.19
24	IAQ	2	1	3	0.29
25	IBP		2	2	0.19
26	ICM		1	1	0.09
27	JAA		1	1	0.09
28	LAY		2	2	0.19
29	MAJ		2	2	0.19
30	NAG	1		1	0.09
31	NAI	268	341	609	59.41
32	NAS	2	1	3	0.29
33	QAO		1	1	0.09
34	TAN	2		2	0.19
35	UAZ	6	18	24	2.34
36	UBA	1		1	0.09
37					
38					

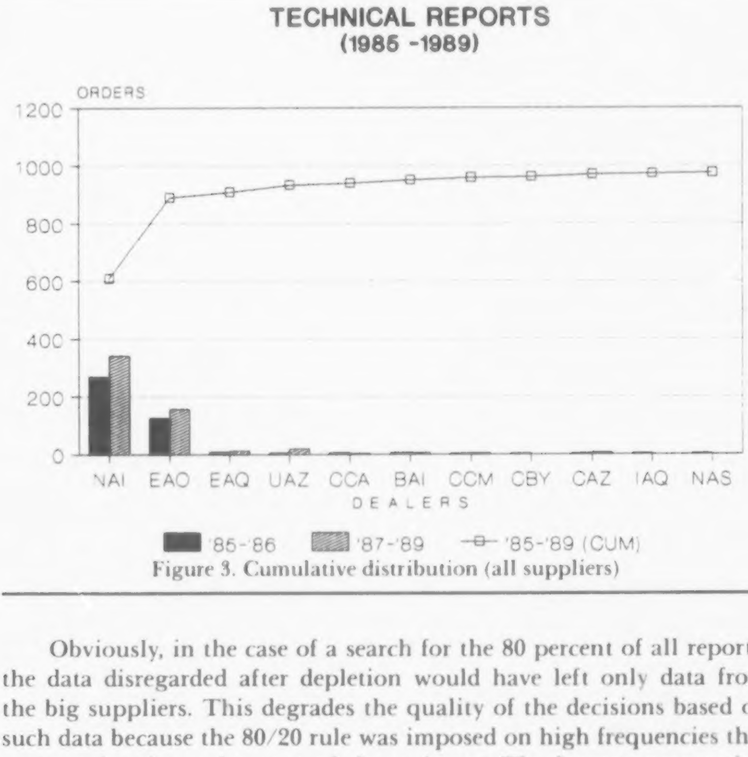


Figure 3. Cumulative distribution (all suppliers)

Obviously, in the case of a search for the 80 percent of all reports, the data disregarded after depletion would have left only data from the big suppliers. This degrades the quality of the decisions based on such data because the 80/20 rule was imposed on high frequencies that account for the major part of the universe. The large concentration of those report producers as represented in the authors' holdings also comply with the so-called "Matthew Effect" (Merton, 1968).

Some useful weighted criteria are:

- Similarity in subject field of the producer
- Potential use in research projects
- Quality and credibility of the source
- History of use
- Bibliographical accessibility
- Availability
- Visibility of originating institution

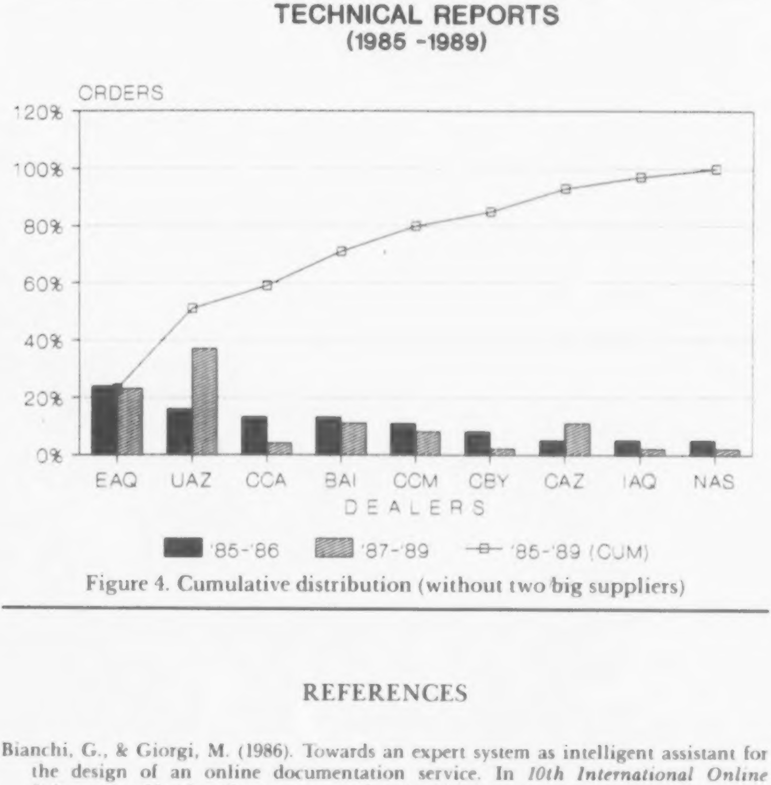


Figure 4. Cumulative distribution (without two big suppliers)

REFERENCES

Bianchi, G., & Giorgi, M. (1986). Towards an expert system as intelligent assistant for the design of an online documentation service. In *10th International Online Information Meeting* (Papers presented at the 10th International Online Information Meeting, 2-4 December 1986) (pp. 199-208). Oxford: Learned Information.

Chachko, A. G., & Stakbovaya, T. M. (1984). Representation of knowledge and decision making in a practical system of artificial intelligence. *Kibernetika, 4* (July-August 1984), 99-108.

Eick, C. F., & Lockemann, P. C. (1985). Acquisition of terminological knowledge using database design techniques. In *Proceedings of ACM-SIGMOD 1985 International Conference on Management of Data, 14*(4), 84-94. Austin, TX: ACM, SIGMOD.

Hayes-Roth, F.; Waterman, D. A.; & Lenat, D. B. (Eds.). (1983). *Building expert systems.* Reading, MA: Addison-Wesley.

Liebowitz, J., & DeSalvo, D. A. (Eds.). (1989). *Structuring expert systems: Domain, design, and development.* Englewood Cliffs, NJ: Yourdon Press.

McGowan, C. L., & Kelly, J. R. (1976). *Top-down structured programming techniques* (2d ed.). New York: Petrocelli/Charter.

Merton, R. K. (1968). The Matthew effect in science. *Science, 159*(3810), 56-63.

Pontigo, J.; Rodriguez, B.; & Ortiz-Gama, S. (in press). Generation of decision rules for an expert system in document supply. In *Artificial intelligence: Expert system and other applications* (Proceedings of the 17th ASIS Mid-Year Meeting, 15-18 May 1988). White Plains, NY: Greenwood Press.

Rodriguez, G., & Rivera, P. (1986). A practical approach to expert systems for safety and diagnostics. *InTech, 33*(7), 53-57.

Waldstein, R. K. (1986). Library—An electronic ordering system. *Information Processing & Management, 22*(1), 39-44.

Weiss, S. M., & Kulikowski, C. A. (1984). *A practical guide to designing expert systems.* Totowa, NJ: Rowman & Allanheld.

Yung-Choa Pan, J. (1984). Qualitative reasoning with deep-level mechanism models for diagnoses of mechanism failures. In R. M. Haralick (Chair), *The First Conference on Artificial Intelligence Applications* (Papers presented at the First Conference on Artificial Intelligence Applications, 5-7 December 1984) (pp. 295-301). Silver Spring, MD: IEEE Computer Society Press.

W. DAVID PENNIMAN*

Libraries and Information Systems Director
AT&T Bell Laboratories
Murray Hill, New Jersey

Walking Your Talk: Why Information Managers are Not High Tech

ABSTRACT

This paper discusses the role of technology in creating successful information services, and also the important role of *people* in creating successful implementations of new technologies. Much has been written about the areas of artificial intelligence and expert systems. This paper will try to stay on a broader level, its ideas applying to a wide range of technologies, well beyond the traditional library arena. Although most readers of these proceedings are concerned with information management within the library context, libraries provide an environment well suited for a more general discussion about change and the role of technology in introducing change necessary for survival.

INTRODUCTION

Libraries are a curious enigma. Librarians have a long history of dealing with change, but in a schizophrenic way. They cling to the past, and yet they are often the heaviest users of technologies, such as computing and telecommunications resources. Some library leaders are noted for their resistance to change, while others are at the forefront of technologically driven innovations. This phenomenon is not restricted to libraries or librarians. Many computer center managers are still clinging to the concept of centralized computing facilities, while others are embracing the reality of decentralized and distributed information processing power.

---

*The author is now President, Council on Library Resources, Washington, DC.

One historical image of the librarian is taken from a novel about a medieval library, *The Name of the Rose*. The passage reads as follows:

> The library was laid out on a plan which has remained obscure to all over the centuries, and which none of the monks is called upon to know. Only the librarian has received the secret, from the librarian who preceded him, and he communicates it, while still alive, to the assistant librarian, so that death will not take him by surprise and rob the community of that knowledge. Only the librarian has, in addition to that knowledge, the right to move through the labyrinth of the books, he alone knows where to find them and where to replace them. He alone is responsible for their safekeeping. (Eco, 1983, p. 37)

Clearly, what is at issue in this image of the library is control. Elsewhere in these proceedings are comments about who should be in control, the system or the user. My question is broader but equally relevant. Who will control access to information, or to information and information technology in a more modern sense? As long as we use control as a measure of success, managers will not embrace technology that diminishes their control.

THE GROWTH OF INFORMATION TECHNOLOGY

We have seen tremendous changes in information technology in our own lifetimes, but from a broader perspective there have been even more profound changes. We have seen storage technology advance from paper to microfilm to magnetic and optical technology. Where we once could store only a few hundred characters per cubic inch, we can now store billions of characters per cubic inch. Transmission capabilities have made similar startling advances. Communications technology has jumped from fifty words per minute of telegraphy to billions of words per minute via glass fibers, and 100 trillion words per minute is within reach. Processing has gone from hundreds to billions of instructions per second, and parallel processing makes the rate practically limitless. Yet our ability to process all this information is virtually unchanged from the time our ancestors emerged from their caves where they had scrawled primitive symbols on the walls. They could process symbols at about 300 units per minute—and so do we. This limit, and our inability to speed up our own processing capacity, is symbolic of our greatest challenge: How to convert all this information being stored, processed, and transmitted into knowledge that is of use to humans.

Strategic Planning: The Challenge to Information Managers

Our ability to use technology to address this last barrier (the barrier to understanding) is sorely limited—not because we lack technological know-how, but because we lack *strategic* know-how. In a book on the Information Age, the following paragraph appears:

> Millions of telephones, thousands of minicomputers, and miles of optical fiber will not create a golden age of. . . information . . . . People always dream about a better future, and our social system encourages this imaginative dreaming. The information society is one such social dream . . . . When discussing a possible better future, we must argue in social, not primarily technological terms. To make that future a reality we have to act in social, not technological terms. (Qvortrup, 1987, p. 134)

When we deal with issues of change, we are dealing with strategic social issues, not technological ones. Paul Strassmann (1985), former vice president of Xerox, agrees. In his book *Information Payoff,* he argues for the pre-eminence of strategy over organizational structure or technology. He states that technology and organization are enablers, but that strategic goal seeking, positioning, and discovery of new "islands" where one can survive are what really make the difference. This calls for a brand of leadership that is opportunistic, entrepreneurial, and able to change direction rapidly with the changing needs of its customers (yes, customers). We all have customers, and sooner or later their satisfaction with our services and their willingness to continue to support us will determine our survival. This is a broader interpretation of what we call user-driven or user-controlled systems. Coming to grips with this inevitable truth—that our survival depends on satisfying our customers—is surprisingly difficult for many organizations that are, in fact, in the service business. Recognizing that those who make funding decisions are one type of customer, while those who receive our service are another type, is important in establishing our customer performance measures and strategies. (Performance measures will be discussed later in this paper.)

Adjectives such as opportunistic, entrepreneurial, or highly flexible do not characterize most information service managers within large institutions, nor do they characterize most library leaders. We have developed our leaders to provide stability and consistency via a centralization of control—to preserve an empire, be it the library, or the computer center, or the management information system. Whenever a new technology threatens to diminish control, it will not succeed on its own merits alone. Pat Battin (1984) has said: "One of the most powerful deterrents to change in conservative institutions . . . is the existence of strong, autonomous, vested interests and the fear of losing one's empire" (p. 170). As long as that empire is measured on the basis of assets controlled (as it is in many libraries, computer centers, or other information service organizations), change will not be embraced, and technology will not be used where it is seen to diminish those assets.

Accountability

The problem is one of motivation, and the problem of motivation rests with how information service providers of today and tomorrow will be measured. For what will they be held accountable? If they continue to be measured on the basis of the size of their stacks, the number of staff reporting to them, or the number of databases under their direct control, there will continue to be static, nonresponsive organizations that fail to serve their customers as fully as they could. The leaders of these services will talk technology, but be thinking about control of assets. These leaders will be skeptical of new information technology, because they are seldom rewarded for increased productivity—especially if it leads to a decrease in their assets, against which their value is judged. So the way we measure success for our information service providers must be changed. With the correct measures, we will encourage them to use technology that holds real promise for drastic re-engineering of their enterprise. F. W. Lancaster (1982) says: "The survival of the library profession depends on its ability and willingness to change its emphasis and image" (pp. 169-70). This author proposes to accomplish that, and to increase the successful use of technological innovation in ways that really—literally—count. Douglas Metzler, in the opening paper in these proceedings, writes that a fundamental change in the library will come from a change of materials in the library. I believe change will come from a more fundamental issue.

A NEW PHILOSOPHY OF INFORMATION LEADERSHIP

Immeasurable vs. Measurable Value Approaches

Information service providers must make a decision. They must choose a new philosophy of information service leadership (Penniman, 1987a). The traditional view is that information organizations are institutions providing service of immeasurable value. Most libraries function under this philosophy. Some MIS facilities do also. Fewer computer centers do, but many are still funded as if they believe this "immeasurable value" philosophy. No commercial information services operate under this philosophy for long. As the overhead costs of information services come under the magnifying glass, this philosophy will cease to be viable.

The alternative philosophy is that every information service/product has a measurable value. The value of a service may be its selection over a competing service when the unit costs of both are made explicit. However it is computed, it needs to be made explicit, or the value may be the lost opportunity cost if the service is not maintained. Charles Fenly suggests, earlier in these proceedings, that an expert system could be used to extend the reach of scarce expert resources and also extend their impact beyond their term of employment. He questions, however, if this benefit could be quantified. Designers of expert systems must address this issue and choose between two approaches.

First, in the immeasurable value approach, information services are justified on qualitative assertions. Resources required are quantified (i.e., budgets), but output measures are de-emphasized (instead, "value" is measured by volumes held or size of budget). The link between mission and output is subjective, and productivity is not (and cannot) be measured. Budgets grow or shrink incrementally (e.g., cut budget by 10 percent) and accountability focuses on resources used. Expenditures on AI projects, for example, must be taken as a matter of faith—not as an investment in the future.

In the second approach, the measured value approach, organizations are justified by quantitative assertions (i.e., improved reference service productivity by 20 percent, provided a return on investment of 35 percent, decreased cataloging expense by 20 percent while holding output constant). Resources required are quantified, but so is output, and productivity is measured. The link between mission and output is objective, and budgets can include individual program values so that decisions are made on the basis of program benefits. Accountability focuses on input *and* output measures. This second approach has serious implications for the infrastructure of an organization. It moves that organization and its services into the mainstream of the broader community in which it resides. It positions the library, for example, as a delivery mechanism rather than a warehouse, with an emphasis on output, not assets. It moves library leaders closer to key decision makers who understand this type of quantification, and closer to MIS and computer center managers. It moves investment on new technologies out of the faith realm. It also increases the potential for power struggles (every benefit has its cost).

Consistent with this second philosophy is the idea that every information service organization should have a clear mission, vision, set of goals, objectives, and strategies to achieve those goals and objectives with measurable results. Metzler writes at the outset of these proceedings that a quality of intelligent organisms is goal seeking and environmental interpretation. Indeed, planning must be part of every intelligent leader's standard operating environment, and technology should be viewed as a key component for achieving the mission and vision of the organization—or it should not be considered.

How the AT&T Library Network Uses the Measurable Value Approach

The vision of the AT&T Library Network, for example, is

> to provide all professional employees throughout AT&T with an electronic window to the vast array of internal and external information services and to assure that the underlying information resources are managed as strategic assets providing a competitive advantage to AT&T. [Our mission is] to provide technical, business, and marketplace information needed by individuals and groups throughout AT&T at competitive cost.

These two concepts together—what we do (our mission) and what we wish to be (our vision)—give us the direction to make choices about appropriate technologies including where we will invest a limited capital budget. Strategic assets, quantification of results, or return on investment are business terms. Libraries must embrace not only such terminology, but also the underlying philosophy of business if we are to survive or, better still, thrive in today's environment. For libraries operate in a competitive environment where scarce resources are allocated by institutional decision makers on the basis of perceived value. Library leaders need not only a dedication to the services they provide, but also a willingness to compete for resources on the same terms as other information-oriented organizations. Computing centers learned long ago to understand their unit costs of their services and to argue in terms of return on investment. Libraries must do the same. This will require librarians to challenge the most fundamental philosophies of leadership in our profession.

Recently, the AT&T Library Network funded a study of the value of its services. This study showed a return on investment of between 400 and 1,000 percent (in line with office automation results, but still so high that many managers don't believe the real leverage of information services). In the area of AI, one could expect the same return on investment for an effective reference support system that reduced the need for on-site reference support, especially with an increasingly dispersed customer base. If one thinks of the return on investment of the technologies discussed in these proceedings, it becomes clear that responding to that challenge is essential.

The Myth of Technological Predestination

Changing the measures of success for information services and service leaders is necessary but not sufficient. Changing their philosophy of management to a more business-oriented one, in which strategic direction and vision play a major role, is also necessary but not sufficient. Major responsibility must also rest with the technology developers. They must recognize that a technology by itself does not succeed.

Little evidence supports the phenomenon of technological predestination. For example, in a study conducted for the National Science Foundation of over 100 information service innovations that failed to reach the marketplace, over 70 percent of the failures were attributed to factors other than technology—factors including management, marketing, and finance (Sweezy & Hopper, 1975). That study, as well as the author's own experience, indicate the need for an activist or interventionist model of change, not one of technological predestination, which is far too passive a view.

A MODEL FOR CHANGE

It is not nearly enough to wait for technology. Both the developers and the embracers of a new technology must understand the conditions for successful use of that technology. Just as there are studies of the failure of innovation, there are also studies of the successful implementation of innovation (Cohen et al., 1979; Dutton & Starbuck, 1979). The findings of these studies point the way and indicate the conditions necessary for success:

1. There must be an understanding of the technology in terms of its advantages over other technologies already available. This understanding must include a thorough knowledge of costs and the relation to processes already in use.
2. Feasibility demonstrations are necessary but not sufficient. Such demonstrations help to identify shortcomings and give early warning signals where improvements are needed.
3. Advocates or champions are needed among both the producers and user groups to assure that early obstacles do not become permanent barriers.
4. External pressures, such as competition and other threats, help to stimulate the implementation process.
5. Joint programs involving multiple organizations provide a broader base of support for the innovation in its early stages.
6. Availability of adequate capital is essential and must not be taken for granted. Ideas do not sell themselves; they require constant attention, and that requires capital.
7. Visibility of consequences is a strong motivator to avoid failure. Announcing publicly an objective makes it more difficult to turn away from that objective.

8. Social support is often a key element and may involve organizations that can provide moral, if not financial, support.
9. Promotional agents, such as the press or other public relations groups, can help to assure that all affected parties understand the technology and how it will benefit them. Such agents also help to elevate the visibility of consequences (see item 7).

These factors lead to a model for change that incorporates two types of bridges between the present and the desired future (Penniman, 1987b). First is a retrospective bridge, or feedback, that compares what we said we wanted with what we have accomplished thus far (i.e., accountability). Second is a forward-acting bridge that is based on intervention, i.e., making the future develop according to our wishes, not someone else's. What ties these two bridges together (accountability and intervention) is an analysis of our successes and failures and a sharing of our experience openly with one another. In the *Journal of the American Society for Information Science (JASIS)*, an article by two other librarians, Lucier and Dooley (1985), states:

> Library administrators have the responsibility to create organizational climates that encourage and promote change. Traditional committee structures are an insufficient approach to anticipate and meet the challenges. Experimentation is essential, improvisation inevitable, and the sharing of both successes and failures a professional and organizational imperative. The great responsibility, however, rests with the individual who must adapt, and adopt the idea of continual change as a goal and a mode of both personal and organizational operation. (p. 47)

We need to learn how to create "learning organizations"; i.e., organizations that treat every effort, every group, every program as an opportunity to share experience and to learn from that experience. That is a challenge for technology developers as well as information service managers. Managers are not high-tech in many cases because they cannot afford to make a mistake, and system developers display a curiously dispassionate view of their systems when things go awry. They are often great at analysis and intervention but fall short on accountability. Managers, on the other hand, are oppressed by the qualitative type of accountability of the past and could actually benefit from the more quantitative analysis that systems designers can offer.

One way system developers can demonstrate a sense of accountability is to demonstrate an in-depth understanding of the full range of conditions necessary for success. They must not only understand those conditions but also be willing to help create them. Having a deep knowledge of the technology is not enough. Developers must also have a deep knowledge of the total environment they are dealing with and the likely conditions that lead to failure or success in that environment.

If the technologists really want to see high-tech information service managers, they must help create them. They must help those managers see how the technology fits into long-range strategic objectives. They must also provide some technologies that support near-term tactical needs (to buy the time necessary for grander plans).

At the 1988 Clinic on Library Applications of Data Processing, *Design and Evaluation of Computer/Human Interfaces* (Siegel, 1991), this author quoted a ten-year-old paper (Penniman, 1979) that argued for system boundaries that recognized the viewpoint of the user—not the systems designer. It was also argued that the "system" boundary should encompass not only the search system and the document delivery system, but also the education system (for users, intermediaries, and designers), the bureaucratic system, and the economic system in which they reside. At the very least (to quote from the 1979 paper), system providers must:

- understand the total system;
- respond to fundamental user requirements;
- use appropriate technology (not necessarily the most advanced technology which, of course, is of most interest to the technologist); and
- establish links with other system components (such as document production systems and document delivery systems).

Now, more than ever, it is essential to establish links with economic and bureaucratic systems with which we must deal. In a report titled "Managing Emerging Information Technology" (Witter, 1986), a similar philosophy is brought home with a checklist that has been modified for this paper. It is presented as a cautionary list, to be considered jointly by technologists and information service managers to avoid the pitfalls that such promising technologies as AI and expert systems might have.

Rules for Failure

To fail, developers or managers need only follow these simple rules:

- Allow too many bright people, who are fascinated with gadgets and removed from the reality of the information service business, to dominate the scene.
- Choose a leader who is very technically oriented and cannot provide a consistent (strategic) focus for his/her staff or cannot communicate well with senior management.
- Operate without clearly defined and measurable performance expectations for either the technical or managerial staff (does that sound like the previously mentioned "measurable value" approach?).

- Ignore or avoid issues such as: cost/benefit analysis, the need for a champion in high places, the impact of the technology on the people in the workplace, or the existing system architecture or other parts of the institution.
- Spend too little time defining the requirements of the information service before selecting one of the latest technologies.
- Ignore the need to weigh trade-offs between choosing a technology now or waiting for better alternatives.

CONCLUSION

It is hoped that this paper has challenged information service managers (and particularly librarians) to be more "business"-oriented in the sense of use of strategies and metrics, and system developers to understand the broader context, in a business sense, in which they are operating. Some reasons have been suggested as to why information service providers may not want to use the latest technologies—even though they may profess great interest in such technology. Finally, some ideas have been presented on where librarians and system designers could work together to avoid some pitfalls.

"Ah," but you say, "I'm not in charge—I'm not the leader. There's little I can do to change how we operate." That is not so; leadership resides anywhere in an organization where there are people with the passion and zeal to take up a vision and to follow that vision to make something happen. The truth is, most effective leaders are servants first (Greenleaf, 1973)—servants to their customers, servants to their institutions, and most important, servants to their vision. If you have a vision of advanced information services in which artificial intelligence and expert systems are a component, and if you want that vision to be a reality, then you must make it happen. You must, therefore, be concerned with the issues raised herein (issues of intervention, analysis, and accountability), and you must "walk your talk" regarding this technology.

REFERENCES

Battin, P. (1984). The library: Center of the restructured university. *College & Research Libraries, 45*(3), 170-176.

Cohen, H.; Keller, S.; & Streeter, D. (1979). The transfer of technology from research to development. *Research Management, 22*(3), 11-17.

Dutton, J. M., & Starbuck, W. H. (1979). Diffusion of an intellectual technology. In K. Krippendorff (Ed.), *Communication and control in society* (pp. 489-511). New York: Gordon and Breach Science Publishers.

Eco, U. (1983). *The name of the rose.* New York: Harcourt, Brace, & Jovanovich.

Greenleaf, R. K. (1973). *The servant as leader.* Newton Center, MA: Robert K. Greenleaf Center.

Lancaster, F. W. (1982). *Libraries and librarians in an age of electronics.* Arlington, VA: Information Resources Press.

Lucier, R. E., & Dooley, J. F. (1985). Cosmology and the changing role of libraries: An analogy and reflections. *Journal of the American Society for Information Science, 36*(1), 44-47.

Penniman, W. D. (1979). Networked databases—A view from the middle. In J. Bichteler (Ed.), *Geoscience information: Publication—processing—management* (Proceedings of the Thirteenth Meeting of the Geoscience Information Society, 22-26 October 1978) (Vol. 9, pp. 39-54). Falls Church, VA: Geoscience Information Society.

Penniman, W. D. (1987a). On their terms—Preparing libraries for a competitive environment. *The Bottom Line, 1*(3), 11-15.

Penniman, W. D. (1987b). Tomorrow's library today. *Special Libraries, 78*(3), 195-205.

Qvortrup, L. (1987). The information age: Ideal and reality. In J. D. Slack & F. Fejes (Eds.), *The ideology of the information age* (pp. 133-145). Norwood, NJ: Ablex.

Siegel, M. A. (Ed.). (1991). *Design and evaluation of computer/human interfaces: Issues for librarians and information scientists* (Papers presented at the 1988 Clinic on Library Applications of Data Processing, 17-19 April 1988). Urbana-Champaign: University of Illinois, Graduate School of Library and Information Science.

Strassmann, P. A. (1985). *Information payoff: The transformation of work in the electronic age.* New York: Free Press.

Sweezy, E. E., & Hopper, J. H. (1975). *Obstacles to innovation in the scientific and technical information services industry* (IPA Monograph 76-2). Washington, DC: Institute for Public Administration. (NTIS No. PB-253 449/3)

Witter, E. D. (1986). *Managing emerging information technology: A decision framework for building competitive advantage* (Business Intelligence Program Rep. No. 738). Menlo Park, CA: SRI International.

CONTRIBUTORS

CHRISTINE L. BORGMAN (M.L.S., University of Pittsburgh; Ph.D. Stanford University) is an Associate Professor in the Graduate School of Library and Information Science at UCLA, and in the communication studies program. Her primary research interests are in information-seeking behavior and in interface design for information retrieval. Her articles have appeared in a variety of information science and communication research journals, and her books include *Effective Online Searching: A Basic Text* (Marcel Dekker, 1984) and *Scholarly Communication and Bibliometrics* (Sage, 1990). She has served on the editorial boards of *Communication Research*, the *Journal of the American Society for Information Science*, the *Annual Review of Information Science and Technology*, and *Interacting with Computers*, and as a member of the Technical Advisory Committee for the Division of Information, Robotics, and Intelligent Systems of the National Science Foundation, and is an elected Fellow of the American Association for the Advancement of Science.

CHARLES FENLY has worked for the Library of Congress since 1974. He has been Assistant Chief of the Descriptive Cataloging Division since 1988. He holds an M.A. degree in Librarianship from the University of Denver and a B.A. in Economics from the University of Texas at Austin.

SUSANNE M. HUMPHREY is an Information Scientist leading the MedIndEx project at the Lister Hill National Center for Biomedical Communications (LHNCBC), a research and development division of the National Library of Medicine (NLM), where she has been developing knowledge-based systems for the past eleven years. The MedIndEx (Medical Indexing Expert) system is a prototype for interactive computer-assisted indexing of biomedical documents in a workstation environment. Previously she had direct experience with NLM's retrieval system since its inception, including indexing, searching, database management, user training, and thesaurus management.

F. W. LANCASTER is Professor of Library and Information Science at the University of Illinois in Urbana-Champaign and Editor of *Library Trends*.

DOUGLAS P. METZLER received his Ph.D. in Cognitive Psychology from the University of California, Davis, in 1981. He was a Visiting Professor of Cognitive Science at Reed College for one year before going, in 1981, to the University of Pittsburgh, where he is currently an Associate Professor of Information Science. He helped establish an interdisciplinary Ph.D. program in Intelligent Systems at the University of Pittsburgh, and a Masters Degree specialization in Artificial Intelligence within the Information Science Department. His research has concerned the development of the Constituent Object Parsing System, and the development of more flexible architectures and modes of communication for expert systems.

SERGIO ORTIZ-GAMA received his B.S. degree in Industrial Engineering in 1986 from the University of Morelos, Mexico, and his B.S. thesis in 1987 was developed at the Mexican Electrical Research Institute (IIE). His thesis work focused on the generation of decision rules for an expert system used in document supply. He joined IIE in 1988, becoming a researcher in the Technical Information Department. He is coauthor of three articles on applications of expert systems techniques to technical information services. His present interests include Information Systems Design and Communication Systems.

JAMES R. PARROTT holds an M.Sc. (Physics) and a B.L.S. from the University of Toronto, and he has written several papers on the theory and practice of reference expert systems.

W. DAVID PENNIMAN is President of the Council on Library Resources, Inc., a not-for-profit operating foundation committed to funding projects that enhance library operations. Before joining the council in 1991, Dr. Penniman served as Director of the Information Services Group, and prior to that, as Libraries and Information Systems Director at AT&T Bell Laboratories from 1984. He has served as Vice-President for Planning and Research for OCLC, where he also established the Research Department in 1978. He has worked as a Research Scholar at the International Institute for Applied Systems Analysis in Austria and as an Information Scientist at the Battelle Memorial Institute in Columbus, Ohio, where he was also Associate Manager of the Information Systems Section, which was responsible for the development of the BASIS online retrieval and data management system. He holds an undergraduate degree in Engineering from the University of Illinois and a Ph.D. in Behavioral Science from the Ohio

State University. Dr. Penniman is active in several professional societies and has published over fifty articles and papers in the areas of information systems research, development, and operation.

YOLANDA I. PLUTE is Product Specialist, Knowledge Access, Mountain View, California. She received her M.L.S. from the University of California, Los Angeles, with a specialization in Information Systems Design. At UCLA, she worked as Research Assistant for Christine Borgman in the area of human-computer interaction. Following graduate school, she held the position of Editor at Dialog Information Services, Palo Alto, California, developing and editing systems documentation and marketing materials.

JAIME PONTIGO-MARTINEZ holds an undergraduate degree in Library Science from Universidad Nacional Autonoma de Mexico and an M.S. from Case Western Reserve University. He completed his Ph.D. in Library Science at the University of Illinois in 1984. He is a member of Beta Phi Mu, Asociacion Mexicana de Bibliotecarios, A.C., Asociacion de Bibliotecarios de Instituciones de Ensenanza Superior y de Investigacion, and Colegio Nacional de Bibliotecarios, A.C., where he served as Vice-President. In addition, he is a member of the American Society for Information Science and has participated in Mexican conferences and has represented Mexico in international conferences. He has served as consultant to the government of Bolivia in 1979, under the sponsorship of the International Development Research Centre, Canada, and the Asociacion de Investigacion Industrial Electica, Madrid, in 1984, in a joint project with Instituto de Investigaciones Electricas, Mexico, where he has been Technical Information Manager since 1981.

GUILLERMO RODRIGUEZ is Professor of Computer Science at the Technological Institute of Monterrey, Morelos, and at the National Center of Technological Research and Development, Mexico. He received his Ph.D. in Computer Science from the University of California in 1981, his M.S. in Electrical Engineering from the Massachusetts Institute of Technology in 1973, and his B.S. in Electrical-Mechanical Engineering from the Technological Institute of Monterrey, Mexico, in 1970. Currently, he is Head of the Information Systems Department at the Mexican Electrical Research Institute. His experience is on analysis and design of computer systems, programming languages, and simulation. He has written about fifty articles and a book on the topics of computer programming development, databases, and artificial intelligence. He formalized the ELKA (entity-link-key-attribute) for the databases conceptual design at Hughes Aircraft Company, California, in 1981. He is a member of the IEEE and of the ACM.

LINDA C. SMITH is Associate Professor in the Graduate School of Library and Information Science at the University of Illinois at Urbana-Champaign. She joined the faculty in 1977 and teaches in the areas of reference, science reference, and online information systems. Her research interests include information retrieval, artificial intelligence, and science information. She is active in a number of professional associations including the American Society for Information Science, the American Association for the Advancement of Science, and the Association for Library and Information Science Education.

EZEQUIEL TOVAR-REYES is Professor of Computer Science at the University of Morelos. He received his B.S. degree in Electrical-Industrial Engineering in 1980 from the Technological School of Morelia, Mexico, and his M.S. in Information Systems from the Technological Institute of Monterrey in 1984. He is a researcher at the Mexican Electrical Research Institute (IIE) in the Technical Information Department since 1980, where he is the Manager of the Automation Area. He has written more than twenty papers on information systems design. He is a member of the IEEE.

IRENE TRAVIS has worked in a wide variety of different types of information applications. She has also been engaged in strategic planning for information systems and technology. Dr. Travis holds an M.L.S. and a Ph.D. in Librarianship from the University of California, Berkeley. From 1972 until 1980, she taught cataloging, classification, thesaurus construction, and library automation at the University of Maryland, College Park. In 1980, she joined Planning Research Corp., McLean, Virginia. Between 1983 and 1988, she was assigned the PRC Government Information Systems R&D group, which was involved with expert systems and natural language processing projects. She left PRC in 1988 to become a member of the Information Policy and Strategy Staff of the Information Technology and Facilities Department of the World Bank. Since 1990, she has been the Section Chief in charge of special projects in the Information Services Division of the bank. In this position, she implements electronic records management programs and manages other records management and archival services. She has published a number of papers in the area of application of artificial intelligence to libraries.

BRIAN C. VICKERY is Professor Emeritus at University College London, and he has acted as Principal Consultant for Tome Associates. His books include *Classification and Indexing in Science, Information Systems,* and *Information Science in Theory and Practice*. He is a member of the editorial board of the *Journal of Documentation*.

AMY J. WARNER is currently Assistant Professor, School of Information and Library Studies, University of Michigan, Ann Arbor, Michigan. She holds an M.A. degree in Linguistics from Indiana University and M.S. and Ph.D. degrees in Library and Information Science from the University of Illinois at Urbana-Champaign. Her research deals with applications of linguistic theory and natural language processing to information retrieval system design. She is the author of "Quantitative and Qualitative Assessments of the Impact of Linguistic Theory on Information Science" and (with Christoph Schwarz and Gerda Ruge) "Effectiveness and Efficiency in Natural Language Processing for Large Amounts of Text," both of which appeared in the *Journal of the American Society for Information Science* in 1991. She has also spoken on linguistics and information retrieval to numerous professional organizations, including the American Society for Information Science, the American Society of Indexers, and the Medical Library Association.

SAMUEL T. WATERS retired as Associate Director of the National Agricultural Library in December 1989. He joined the NAL staff in November 1970 as Deputy Director for Resource Development and became Associate Director in 1974. During his tenure at NAL, he emphasized the application of new technologies to traditional library operations and expansion of the knowledge center concept of the library. Before joining NAL, Mr. Waters served as Chief of the Reference Services Division and then as Deputy Associate Director for Library Operations at the National Library of Medicine. He began his library career in clerical and technical positions at the Library of Congress while earning his M.S.L.S. at Catholic University. He held professional positions at LC and several academic libraries and rose to become Associate Director of Georgetown University Libraries before he returned to the government at NLM. Mr. Waters did his undergraduate work at the University of Colorado where he earned his B.A. in philosophy and general studies, summa cum laude, and was inducted into Phi Beta Kappa.

STUART L. WEIBEL has been in the OCLC Office of Research since 1985. During that time, he has managed projects in automated cataloging, optical character recognition, and electronic publishing. Dr. Weibel earned his Ph.D. in Pharmacology from the Ohio State University.

INDEX

1st CLASS, 164, 168, 172
80/20 rule: and expert system efficiency, 60, 232, 234-235, 262-264

ADAPT. *See* Project ADAPT
AGRICOLA (AGRICultural OnLine Access), 163, 165, 166
AGRICOLearn, 163, 164
Ahlswede, T., 206
AIMS. *See* Automated Indexing and Management System
Allen, J., 199, 227
Allen, J. F., 27, 200
Anderson, J. R., 34
Andre, P. Q. J., 78
Anglo-American Cataloguing Rules, 2d ed. (AACR2), 57, 70, 71, 115, 234
Answerman, 140, 149, 155, 165
Apple Library Tour, 130, 141
Aquaculture expert system. *See* AquaRef
Aquaculture Information Center (AIC), 166
AquaRef, 140, 149, 155, 165-166
Aquatic Sciences and Fisheries Abstracts (ASFA), 166, 172
Arshad, F., 182
Artificial intelligence: current issues in, 15; dialectics of, 4; future directions, 36-39; learning, 5; practicality, 44-45; problem solving, 4; research areas, 44; response to the information explosion, 9; simulating human thought, 4. *See also* Expert systems; Intelligence; Knowledge bases; Learning; Natural language understanding; Reasoning
Ashley, K. D., 22
AT&T Library Network, 272
Automated cataloging. *See* Cataloging; Computer-assisted cataloging
Automated indexing. *See* Indexing, automated
Automated Indexing and Management System (AIMS), 116
Automated Title Page Cataloging Project, 71-72

Backward chaining search, 123, 133, 222-223
Bailey, Charles, 119
Barthes, C., 243
Batch processing, 111
Bates, Marcia J., 141, 147, 150, 189,
Bates tactics, 146-154
Battin, P., 10, 269
Beaugrande, R. A. de, 210
Beerel, A. C., 53, 65
Belkin, N. J., 179, 182, 183, 186, 210
Bender, A., 78
Benson, J. A., 203
Berwick, R., 200
Bhandaru, N., 208
Bianchi, G., 255
Bielawski, Larry, 168
Birnbaum, L., 199
Biswas, G., 206
Bivins, K. T., 129, 139
Blackboard systems, 28, 186
Blair, D. C., 205, 209
Boguraev, B., 209
Borgman, Christine L., 180, 181, 189
Borko, Harold, 71
Boyce, B. R., 202
Brajnik, G., 207
Briscoe, T., 209
Brooks, H. M., 140, 179, 180, 182, 183; 194, 210
Brown, J., 22
Buchanan, B. G., 216
Burger, Robert, 72
Butkovitch, N. J., 141
Butler, R. W., 163
Byrd, R. J., 209

CAI. *See* Computer-assisted instruction
Canonical user models. *See* Static user models
Carbonell, J. G., 199
Carnahan, Paul, 141
Carnegie-Mellon University Libraries, 73

Case-based reasoning, 21-22
Cataloger's Assistant, 73
Cataloging: automation research, 70-78; benefit of automation research, 72; nontraditional, 74; patron's perspective, 68; works, 69. *See also* Computer-assisted cataloging; "Virtual" catalog
Cataloging workstation, 73
CATALYST, 57-58
CatTutor, 172
CEDOCAR, 243
Cerf, V., 6, 8
Ceri, S., 39
Chachko, A. G., 256
Charniak, E., 15, 29, 31
Chase, Stephanie, 167
Chien, D., 87
Chisman, J., 140
Chomsky, N., 204
Circumscription. *See* Nonmonotonic reasoning
CIRT, 242-243
Clancey, W. J., 155, 227
Clemencin, G., 245
CLOS (Common Lisp Object System), 115
Closed-world assumption, 19
Clowes, I., 182, 184
Cognitec, 240
Cohen, H., 273
Cohn, A. G., 23
Cole, I., 182
Common Lisp, 115
Common sense reasoning, 16-24, 226-227
Communication, model of, 182
Computer-assisted cataloging, 72-74; Duplicate Detection Project, 73; fully automatic cataloging, 70-72
Computer-assisted instruction (CAI), 155, 172. *See also* Intelligent tutoring systems
Conceptual models, 181
Connectionism, 35-36; neural networks, 173
Cooper, W. S., 232
Coordinate indexing. *See* Indexing, coordinate
Croft, W. B., 204, 206, 207, 208, 210
Crystal, D., 199
Cyc Project, 32, 226, 228-236; implications for libraries, 236
CycL., 229-230

Dahlberg, I., 229
Daniels, P. J., 179, 181, 182, 183, 184
Das-Gupta, P., 207
Data-driven reasoning. *See* Forward chaining search
Databases: compared with rule bases, 120-122; kinds of query responses, 120
DATASTAR, 242, 243
Davidson, Lloyd, 119
Davies, R., 57, 70-71
Davis, D. B., 210
Davis, R., 24
Dede, C., 155
Default logic, 18, 20, 91
DeJong, G. F., 35
de Kleer, J., 22
De Mey, M., 180
Dependencies, 99
DeSalvo, D. A., 255, 256, 257
Descriptive cataloging expert systems, 56-58; CATALYST, 57-58
Dewey Decimal Classification (DDC), 230; automatic selection, 115
Dewey, M., 230
Dillon, M., 204, 206, 207
Diskin, G. M., 130, 140
DISNET, 240, 252
Document delivery expert system. *See* SEADO
Dooley, J. F., 274
Doszkocs, T. E., 201
Doyle, J., 18, 41
Dressler, W. U., 210
Duke, John, 69
Duplicate Detection Project, 73
Durfee, E. H., 28
Dutton, J. M., 273
Dynamic user models, 183-184

Eaton, N. L., 78
ECHO, 252
Economic Research Service (ERS) USDA, 168
Eick, C. F., 256
ELAND, 39-40
Electronic bulletin boards: ALF (Agricultural Library Forum), 167-168
Electronic collections, 6-8
Endres-Niggemeyer, B., 78

Entailment, 13-15

Ercegovac, Zorana, 71

Eriksson, L., 129, 139

ERLI/MINITEL, 245-246, 252

Ertel, M., 130, 141

ESP-Advisor, 57

Etherington, D. W., 21

EURISKO, 243-245, 247, 252

Expert System for Document Supply. *See* SEADO

Expert system shells: definition of, 216; limitations of, 115; compared with database management systems, 216. *See also* 1st CLASS; ESP-Advisor; EXSYS

Expert systems: costs, 64-65; current capabilities, 36-37; data structures, 41; design requirements, 37-40; development issues, 52-56, 255-256; domain suitability criteria, 54-55, 58; explanations, 41, 42; for descriptive cataloging, 56-58; for document delivery, 254-266; for indexing, 81-117; for ready reference, 164-166; for reference, 118-160, for series work, 59-61, for shelflisting, 61-64; Library of Congress (LC) study, 58; organizational suitability criteria, 55-56; potential benefits, 53; suitability for library technical services, 50-52; training in development of, 166-167

Explanations. *See* Expert systems, explanations

Explicit user models. *See* Stated user models

EXSYS, 258

Facets. *See* Frame-based models, facets

Fadell, J., 130, 141

Falkenhainer, B., 34

Fault tree analysis (FTA), 257

Fenichel, C. H., 189

Fenly, Charles, 53, 58

Fidel, R., 189, 203

Fikes, R. E., 25

Finders, 168

Finin, T. W., 197

Fishery Information Service at the Food and Agriculture Organization of the UN, 171

FNIC-AID (Food and Nutrition Information Center-Artificial Intelligence Demonstration), 166

Fong, S., 200

Forbus, K. D., 22

Forward chaining search, 123, 133, 223-224

Fox, Martha, 70

Frame-based models, 12, 40, 87-104, 113; facets, 96, 104

Freeman, Robert, 171

Friedman, C., 200

Full-text searching, 169-171

GenBank, 116

Gentner, D., 23, 34

Georgeff, M. P., 28

Gibb, F., 57, 58, 78

Gilbert, G. N., 182, 184

Ginsberg, M. L., 18

Giorgi, M., 255

Glize, P., 243

Gorman, M., 57, 234

Gray, A. S., 204, 206

Greenleaf, R. N., 276

Greiner, R., 34

Grosz, B. J., 200

GRUNDY system, 184-185

Guha, R. V., 32, 229, 230, 231, 232, 235

Haas, S. W., 206

Halasz, F., 181

Halpern, J., 241

Hanfman, D., 140, 149, 155, 166

Harmon, P., 157

Harris, H., 53, 58

Harris, L. R., 210

Harris, M. D., 200

Hasling, D. W., 30

Hayes, P., 25, 196, 199

Hayes-Roth, B., 28

Hayes-Roth, F., 28, 256

HDTV. *See* High-definition TV

Hernon, P., 165

High-definition TV (HDTV), 9

Hjerppe, R., 58

Hocker, Susan, 167

Hollnagel, E., 55

Hooper's Indexing Consistency Measurement, 86

Hopper, J. H., 273

Horn clause formalism, 19

Horsfield, Jim, 168

Horty, J. F., 21

Humphrey, Susanne M., 87, 221, 232

Hutton, S. M., 130, 139
Hypertext systems, 120, 164
HyperTIES, 164
Hypothesis-driven reasoning. *See* Backward chaining search

IMIS, 247-248, 252
Implicit user models. *See* Inferred user models
IMSATT, 163
Indexing, automated: 81-177; coordinate, 82, 83; hierarchies, 84; knowledge-based, 87-104; multiplicity, 83; precoordination, 83; specificity, 82; variability, 85, 232
Indexing Aid Project, 232. *See also* MedIndEx system
Indexing workstation, 84
Individual user models. *See* Dynamic user models
Inference, 14, 19; explanations, 142. *See also* Backward chaining search; Forward chaining search
Inferred user models, 184-185
Informal models, 21
Information access, control of, 267-268
Information Function (IF), 130, 140
Information Machine, 130, 141
Information retrieval, characteristics of, 188-189
Information retrieval systems: ambiguity and synonymy in, 203; redundant answers, 120-121; comparison of experimental and operational systems, 205-207. *See also* Reference expert systems; Full-text searching
Information services management: accountability, 270; changing measures of success, 272-273; immeasurable value approach, 271; leadership, 269, 270-271; measurable value approach, 271; strategic planning, 269
Information technology, growth of, 268
Ingrand, F. F., 28
Inheritance, 21, 40, 83, 90, 100-104, 221-222
INSERM interface, 241-242
Institut National de la Sante et de la Recherche Medicale. *See* INSERM
Instituto de Investigaciones Electricas (IIE), 257
Intelligence: definitions, 3-4, 9-13, 16-24, 29, 31; entailment, 13-15
Intelligent computer-assisted instruction (ICAI), 155. *See also* Computer-assisted instruction
Intelligent interfaces. *See* Interfaces, intelligent
Intelligent multilingual interface (IMIS), 240, 241
Intelligent tutoring systems (ITS), 24, 142, 155-156. *See also* Computer-assisted instruction
Interfaces, intelligent, 239-253; functions of, 240-241; problems with construction of, 248-252. *See also* CIRT; ERLI/MINITEL; EURISKO; IMIS; INSERM interface
International Agricultural Research Centers, 170
IR-NLI-II system, 207
ISLAND, 39-42
ITS. *See* Intelligent tutoring systems

James, B., 70
Johnson-Laird, P. N., 23
Jorgensen, C. L., 206

Kang, K. C., 55
Kapoor, A., 221
Kass, A., 29
Katz, William, 128, 131
Kay, M., 204
Kelly, J. R., 257
King, D., 157
King, J. A., 22
Knowledge, deep compared with surface, 156-158
Knowledge acquisition, 32; verification, 111
Knowledge base management systems, 87, 104-113; consistency maintenance, 104; knowledge acquisition, 104-111; LISP editor; 106; syntax checking, 104-113
Knowledge bases: 80/20 rule, 234-235; definition of, 216; general knowledge, 225-226; problems with large size, 231-232; paradox of structure, 233-234; updating, 235-236; variability, 232; qualitative models, 226-227; quality of, 224-

228. *See also* Cyc Project
KnowledgePro, 171
Knowledge representation, 215-238; frames, 221; inference, 222-224; object-oriented approach, 220-222; objects, 217; problem areas, 224-228; procedurally based systems compared with object-based systems, 217; procedurally oriented knowledge structures, 218-220; relational analysis, 217-218; rules, 217; scripts, 222, 227; slots, 221; specific methods of, 218-222; weights, 223-224
Knowledge Representation Hypothesis, 11-14
Kobsa, A., 200
Koll, M. B., 203
Kolodner, J. L., 22, 34
Krawchuk, B. J., 35
Krovetz, R., 207
Kuipers, B., 22
Kulikowski, C. A., 256
Kulkarni, D., 34

Laird, J. E., 34
Lancaster, F. W., 203, 204, 208, 270
Law of Diminishing Returns. *See* 80/20 rule
Learning, 31-36; active discovery learning, 34; connectionism, 35-36; explanation-based learning, 34-35; failure driven learning, 34; probability-based approach, 262; systems, 33; weighted-based scheme, 262; by analogy, 33-34; problem solving, 34
Lebowitz, M., 225
Lenat, D. B., 32, 34, 229, 230, 231, 232, 235
Lesk, M., 209
Lesser, V. R., 28
Lev, Yvonne, 168
Levy, L. S., 55
Lewand, R., 168
Lewis, C., 35
Lewis, D. D., 204, 206, 208
Library of Congress (LC): expert systems feasibility study, 53; preservation of materials project, 9-10; shelflisting procedures, 61-62; training program in series work, 59
Library of the future, 5-11; campus computing center as prototype, 8-9; electronic collections, 6-8; electronic technologies support, 8; library applications of artificial intelligence, 5-6; physical characteristics, 8-9; social function, 9; traditional materials, 8
Liddy, E., 206
Liebowitz, J., 255, 256, 257
Ling-Hwey Jeng, 71
Linguistic String Project (LSP), 200
Linguistics: and natural language processing, 195-196, 207
LISP, 115, 164, 217
Lister Hill National Center for Biomedical Communication (LHNCBC), 116
Litman, D. J., 200
Local area networks, design tools, 39, 40
Lockemann, P. C., 256
Long-term user models, 185
Lucier, R. E., 274
Lucky, R., 9

Mace, S., 171
Machine learning, 228. *See also* Learning; Cyc Project
Map cataloging, 71
MARC format, 62, 226
Marcus, M. P., 209
Maron, M. E., 205, 209
Marsh, E., 200
Masterman, M., 203
Matthew Effect, 264
McCalla, G. I., 155
McCarthy, J., 20, 25
McClure, C. R., 165
McCray, A. T., 206, 207
McDermott, D., 27
McDermott, John, 44
McDonald, L. K., 207
McGill, M. J., 204, 205
McGowan, C. L., 257
McKinin, E. J., 209
MedIndEx system, 81-117; facets, 87; data structure, 87; Indexer Interface, 85-86; Journal Assignment Utility, 85; MedKB Manager, 87, 104-113; MeSH Indices Report Generator, 86; user interface, 87-104
MEDLARS, 82
MEDLINE, 82, 86, 87, 97, 99, 113, 241,

242
MEDSTATS, 168
Mental models, 23-24, 181
Meredith, J. C., 139
Merton, R. K., 264
MeSH (*Medical Subject Headings*), 82, 116, 241, 242
Metathesaurus, 116
Metzler, Douglas P., 12, 13, 36, 37, 39, 206, 270, 271
Michalak, T. J., 130, 140
Microelectronics and Computer Corporation (MCC), 228-229
Miller, N. E., 87
MINITEL, 241
Mishkoff, H. C., 197
Mitchell, T. M., 33
MITI, 240
Model 204 database management system, 116
Model-based reasoning, 23-24
Molto, Mavis, 71
Monotonic logic, 17
Montgomery, C. A., 204
Mooney, R., 35
Moore, J. D., 30
Moran, T., 181
Moravec, H., 176
Morgan, J. L., 210
Moyne, J. A., 204
Multifunctional software, 174
Multimedia products, 174-175
Multiplicity, 83
MYCIN, 30, 216
Myers, J. E., 130, 141

National Agricultural Library (NAL): Advanced Technology Demonstration Center, 164; Animal Welfare Information Center (AWIC), 171; background, 162; expert systems development, 161-177; future of new technologies, 173-176; networking, 162; videodisk technology, 163, 169
National Agricultural Text Digitizing Project (NATDP), 170; Personal Librarian, 171
National Air and Space Museum, 170
National CD-ROM Sampler, 170
National Center for Biotechnology Information (NCBI), 116
National Library of Medicine (NLM), 232; MEDSTATS, 168; Unified Medical Language System (UMLS), 116, 209. *See also* MedIndEx system
Natural language processing, 194-214; and experimental information retrieval, 203-207; compared with computational linguistics, 195-196; compared with natural language understanding, 196; development issues, 199-201; future of, 210; information retrieval system parallels, 201-203; lexicons, 198, 251; linguistic levels, 198-199; morphological analyzer, 198; parser, 198; pragmatic analyzer, 198; query modification, 251-252; semantic analyzer, 198; subject scope, 251; system architectures, 197-199; transformational-generative grammar, 204; user interfaces, 205. *See also* Interfaces, intelligent
Natural language understanding, 10-11, 28-31, 196; context recognition, 31; explanation-based understanding, 30; plan-based understanding, 29
Natural Language Systems Project. *See* SPECIALIST
Needham, R. M., 203
Neural networks. *See* Connectionism
Newcomb, T. M., 182
Newell, A., 12
Newton Database server, 75
Nielsen, N. R., 158
Nilsson, N. J., 25
NLM. *See* National Library of Medicine (NLM)
Nonmonotonic reasoning, 16-21
Nonrepresentational systems. *See* Connectionism
Nontraditional cataloging, 74
Norman, D. A., 24, 181
Norvig, P., 31

Object-based systems, 217
OCLC: Online Union Catalog, 72, 73; research projects, 71-74
OCR. *See* Optical character recognition
Oddy, R. N., 180, 210
Olsen, P. R., 53
O'Neill, E. T., 73
Online Reference System (ORS), 140

Optical character recognition (OCR), 72, 77, 78, 169-170
ORA (Online Reference Assistance), 130, 140
Oros, J., 130, 141
Otano, Hernan, 170

Palmer, R. C., 129
Pareto's Law of Diminishing Returns. *See* 80/20 rule
Parrott, James R., 119, 130, 140, 146, 150, 152, 156
Patrias, Karen, 168
Peachey, D. R., 155
Penniman, W. David, 189, 270, 274, 275
Personal Librarian, 171
Physical Symbol System Hypothesis, 12, 14
Pisa, M., 168
Planning, 25-28, embedded systems, 28; multi-agent, 28; opportunistic, 28; problem decomposition approaches, 26
PLEXUS, 140, 151, 154, 185
POINTER, 140
Pontigo, J., 261
PRECIS, 218, 220
Precoordination. *See* Indexing, precoordination
Prerau, D. S., 54
Problem solving, 4
Project ADAPT, 75-77
Prolog, 19, 115, 164
Putz, W., 78

Qualitative reasoning, 22-23
Question-answering systems. *See* Reference expert systems
Qvortrup, L., 269

R1, 44
Rafats, J., 167
Ready reference expert systems, 164-166. *See also* Reference expert systems
Reasoning, 16-28. *See also* Case-based reasoning; Common sense reasoning; Informal models; Model-based reasoning; Nonmonotonic reasoning; Temporal reasoning; Qualitative reasoning
Reference and Information Station, 130, 139
Reference expert systems, 118-160; backward chaining in, 123; definition, 118; forward chaining in, 123; future of, 158-159; indexing attributes in, 131, 134, 136; precision in, 126, 128-129, 131, 136; realistic weights in, 123-124, 126-129, 131, 132, 137; structures with single-stage retrieval, 122-130; structures with multi-stage retrieval, 130-154; updating in, 124-129, 131, 134, 137. *See also* Answerman; Apple Library Tour; AquaRef; Information Function; Information Machine; Online Reference System (ORS); ORA; PLEXUS; POINTER; REFLES; REFLINK; REFSEARCH; REFSIM; Technical Writing Assistant
Reference process, types of answers, 119-120
REFLES, 129, 139
REFLINK, 129, 139
REFSEARCH, 139
REFSIM, 146, 150, 152, 154, 156
REGIS (REGional Information System), 172
Reiter, R., 19, 20, 24
Relational database management systems, 113-114
Reyes, C., 236
Rich, E., 182, 183, 184, 185
Rissland, E. L., 22
Rivera, P., 257
Robb, D. W., 169
Robertson, Craig A., 167
Robertson, S. E., 242
Rodriguez, G., 257
Rolston, D. W., 60
Rovira, C., 236
Rule bases, 12-13; compared with databases, 120-122
Rule trace, 30

Sager, N., 200
Sahlu, Marcus, 169
Salton, G., 204, 205, 206, 207
Sargeant, H. A., 241
Savvy, 173-174
Schank, R. C., 34, 199
Schmierer, Helen, 77
Schneider, W., 35
Schwarz, C., 204, 206

Schwarz, Helga, 71
Scientific-numeric databases, 175
SEADO: architecture, 256-261; learning capabilities, 261-264
Sears List of Subject Headings, 236
Second-Order Platonism, 13, 14
Seeger, T., 186
Semantic analysis, 20[illegible]
Semidecidable systems, 18
Series expert systems: domain suitability criteria, 59-60; potential benefits of, 61
Sharif, C., 57, 58
Sharif, C. A. Y., 115
Sharples, M., 217
Shelflisting, cuttering, 61-62
Shelflisting expert systems: domain suitability criteria, 62-63; potential benefits, 64
Shoham, Y., 27
Short-term user models, 185
Shortliffe, E. H., 30, 216
Siegel, M. A., 275
Sievert, M. E., 202, 209
Simon, H. A., 12, 34, 45
Situation calculus, 25-26
Skalak, D. B., 22
Smith, B. C., 12
Smith, D. E., 130, 139
Smith, Karen F., 140
Smith, R. G., 216
SOAR system, 34
Soergel, D., 189
Softex GmbH, 247
Sparck Jones, 179, 189, 200, 203, 204, 206, 207
SPECIALIST, 116
SQL (Search Query Language), 113-114
Stakbovaya, T. M., 256
Starbuck, W. H., 273
State space search, 25
Stated user models, 184
Static user models, 183
Stereotypical user models, 183
Stevens, A. L., 23
Stillings, N. A., 183
Strassman, Paul, 269
Structured flow charts (SFC), 257
Subject heading correction, 73
SunOS, 115
SunSPARCstation, 115
SunView, 115
Svenonius, Elaine, 71
Swab, J., 167
Swartout, W. R., 30
Sweezy, E. E., 273
Sycara, K., 28
Symbolic representation, 11-16

Tait, J. I., 204, 206, 207
Talmy, L., 29
Tanca, L., 39
Taylor, L., 200
Technical Writing Assistant, 141
TELESYSTEMES-QUESTEL, 241, 243
Templates. *See* Frame-based models
Temporal reasoning, 27
Text digitization projects. *See* Optical character recognition
The Interactive Encyclopedia System (TIES), 164
Thoma, G. R., 78
Thomas, S. E., 172
THOMAS, 180
Thomason, R. H., 21
Thompson, C. L., 242
Thompson, R. H., 207
Thompson, W. B., 30
Thought processes, models of, 180-183
Tome Associates, 247
TOME SEARCHER, 171, 203, 247
TOME Selector, 247
Touretsky, D. S., 21
Transformational-generative grammar, 204
Trautman, R., 141, 150, 151, 154
Treat, W., 140
Truth maintenance systems, 18, 41
Tulving, E., 22
Tuttle, M. S., 209

UMLS. *See* National Library of Medicine (NLM), Unified Medical Language System
Uncertainty. *See* Nonmonotonic reasoning
University Paul Sabatier, 247
User interface, 84; knowledge acquisition, 106-111; natural language processing techniques, 205. *See also* Interfaces; Cataloging workstation; Indexing workstation; MedIndEx
User modeling, 154
User models, 178-193; applications to information retrieval, 188-190; cate-

gorization criteria, 182-185; characteristics included in, 182; choice of, 187-188; compared with expert systems, 178-179; compared with conceptual models and mental models, 181-182; dynamic, 183-184; example of static model, 191; facets, 183; GRUNDY system, 184-185; inferred, 184-185; intelligent interfaces, 179; long-term, 185; PLEXUS, 185; problem description models, 185-186; research, 179-180; short-term, 185; stated, 184; static, 183; stereotypical, 183; THOMAS, 180; triggers, 183; types of, 180-182, 186

Vickery, B. C., 140, 185, 240, 247
"Virtual" catalog, 69
Vizine-Goetz, D., 73
von Flittner, S., 141, 150, 151, 154

Wahlster, W., 200
Waldstein, E. K., 255
Walker, D. E., 226
Wallis, J. W., 30
Walters, J. R., 158
Warner, A. J., 196, 199
Waterman, D. A., 53, 55
Waters, Samuel T., 140, 149, 165
Weibel, Stuart, 71-72, 78
Weinberg, B. H., 203
Weischedel, R. M., 199
Weiss, S. M., 256
Wersig, G., 186
Weston, C. V., 172
White, Howard, 139
Wick, M. R., 30
Wilensky, R., 29
Williams, J., 39
Williams, M. E., 205
Wilson, Patrick, 69
Winkler, P.-W., 57, 234
Winograd, T., 198
Winston, P. H., 33
Witten, I. H., 35
Witter, E. D., 275
Works. *See* Cataloging, works
Workstations. *See* Cataloging workstation; Indexing workstation
World knowledge base, 15

XCON. *See* R1

X Window System, 115

Yung-Choa Pan, 256

Zidar, J. A., 78, 170